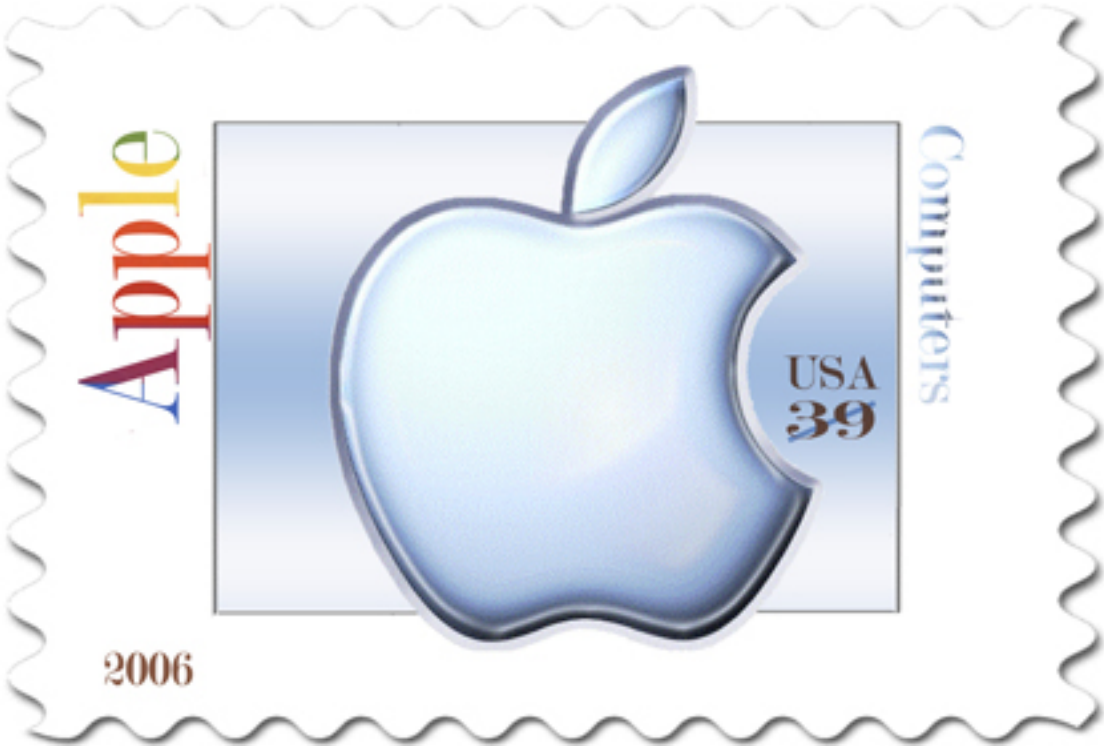
ATPM

12.07 / July 2006 Volume 12, Number 7



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

Cover Art

 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
	Vacant
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
	David Ozab
	Sylvester Roque
	Charles Ross
	Mark Tennent
	Evan Trent
	Vacant

Artwork & Design

Layout and Design	Michael Tsai
Web Design	Simon Griffee
Cartoonist	Matt Johnson
Blue Apple Icon Designs	Mark Robinson
Other Art	RD Novo
Graphics Director	*Vacant*

Emeritus

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

Contributors

Matthew Glidden, Ted Goranson, Matt Johnson, Miraz Jordan, John Lowrey, Robert Paul Leitao, Wes Meltzer, Sylvester Roque, Charles Ross, Mark Tennent, Michael Tsai, *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. ISSN: 1093-2909.

Production Tools

Apache, AppleScript, BBEdit, Cocoa, Docutils, DropDMG, FileMaker Pro, GraphicConverter, LaTeX, Mesh, make, Mailman, Mojo Mail, MySQL, Perl, Photoshop Elements, PyObjC, Python, rsync, Snapz Pro X, ssh, Subversion, Super Get Info.

Reprints

Legal Stuff

• • •

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 July issue of *About This Particular Macintosh!* Hot summer weather covers much of the nation, as residents of the mid-Atlantic states are happy to see star-spangled evening skies after several days of continuous rain. Floods in the east and many wildfires in the west have made for an exciting start to the summer season. In a few days, the United Sates will celebrate the anniversary of the country's Declaration of Independence. Each month we bring you an independent look at the state of the Mac. Come rain, fire, heat, or snow we offer the latest news and freshest reviews in an easy to read monthly format.

France

Our nation's oldest ally is currently the iPod Nation's most formidable foe. At press time, the French legislature adopted measures to compel Apple Computer to open its iTunes Music Store to music players from other manufacturers. More exactly, the legislation seeks to force Apple to sell songs in France that can be played on digital music players other than the iPod. That's the news headline. How the law actually works following key amendments to water-down its impact may be another story. Apple has made hints the company will withdraw its music sales from the nation of France if such new laws threaten the iPod Nation's dominance. It may take months before the implications of the new laws are determined.

Options and Fewer Options

Until recently, stock options have been a big component of tech sector executive compensation. Recently enacted regulations now require enterprises to "expense" the value of stock options–based compensation. Long shrouded from public view, stock options and the manner in which option dates are determined are among the latest high-profile scandals to hit corporate America.

Here's an overly simplified explanation of the matter: executives are granted the right to purchase their employer's stock at a pre-set price. The difference between the actual trading price on the day the options are exercised or used and the pre-set option price is taxable income to the executive and represents stock-based compensation. The pre-set option price is usually set on the day the options are granted. For years, the difference between the price paid by the executive for the shares at the pre-set option price and the dollars the company might have brought into its own coffers had the shares been sold at market price never appeared as a "cost" or "expense" on the company's books. Now that the regulations have changed and the cost must be reported, fewer companies are issuing options, and companies who continue to grant them are granting fewer options.

The question now is: who selects the date the options are granted and how is the date selected? It seems rather strange that so many companies were able to select as the option date for their executive option grants the exact day (or close to it) during a fiscal quarter

their company's shares were trading at their lowest price, thus increasing the income for the executives when the options were exercised or used.

Before trading began on the last day of June, Apple announced the company had discovered some "irregularities" in the manner in which option prices were selected. Most analysts see Apple's disclosure as minor compared to the actions of companies at which options had been aggressively abused. Companies now have fewer options when it comes to granting options.

Shake It Up, Baby

Apple Computer recently announced the release of Shake 4.1 at a dramatically reduced price of $499 for Mac users. This represents a significant reduction from the price of Shake 4 at $2,999. The latest version of the popular professional compositing product is also a Universal Binary application. This means it will run natively on both PowerPC- and Intel-based Macs. At the new price, Shake 4.1 will shake up the industry by providing access to one of the best compositing tools available at a cost many amateur moviemakers can afford.

New Products

Summertime is here, and with the warmer weather comes hot rumors of new Apple products. We expect to see new Intel-based Macintosh minitowers before Labor Day and new iPods in time for the Christmas shopping season.

There's much talk of an iTunes movie store. Should Apple venture into online movie distribution, watch for subtle hints of new convergence products. We suspect Apple will leverage its leadership in digital content distribution with new solutions for the home.

Update Your Mac as We Update Your News

We offer a suggestion: each time the latest issue of *About This Particular Macintosh* arrives in your inbox, make it time to use the Mac's Software Update utility. You'll enjoy each issue of ATPM even more if your Macintosh is as up-to-date as each of our monthly issues. Welcome to summer.

Our July issue includes:

Bloggable: It's Time to Say Goodbye

A prominent Mac user switches to Ubuntu Linux on a Lenovo PC, and sends the chattering classes atwitter. Plus, OS X kernel, *Apple v. Does*, and more.

MacMuser: How do I love Call of Duty? Let me count the ways.

Two-fingers salute to a couple of newcomers.

MacMuser: Wherefore Art Now, Jonathan?

A beige box is a computer standard that implies unremarkable specifications and questionable reliability. Beige is also the most popular shade of pantyhose. Therefore beige boxes are pants.

Outliners: Outlining Interface Futures

Ted Goranson's *ATPO* does something a little different this month. It explores some unusual notions of outlining, and suggests futures for user interfaces.

FileMaking: Script Parameters and Results

This month, *FileMaking* continues its look at scripting capabilities.

Web Accessibility: The Clayton's Web

"Once you understand the Visitor, it's easy to notice that some visitors will *see* what's on your Web site, while many will *see and interpret* only the coding behind it. And that's why the coding is so important. Which is why Apple's iWeb is such very disappointing software."

How To: Maybe You Ought to Be Using Automator

Sylvester Roque makes good on his resolution to learn Automator, and takes us along for the lesson.

Desktop Pictures: Alaska

John Lowrey of Northern Softworks provides this month's desktop photos from Alaska.

Cortland

This month's Cortland features a radical departure as artist Matt Johnson explores a corner of the Web comics universe.

Review: DiscBlaze 6.1.6

If the faster burning speeds and customization are of importance to you, DiscBlaze may be well worth the currently reduced price. Charles Ross, however, found that his simple process of using Disk Utility still suits him best.

Review: Dobry Backuper 1.5

Backuper seems to be in need of some maturing, though it does work as advertised, especially if you're backing up to a hard drive. Charles Ross' advice: make use of the 30-day trial to see if it suits your needs.

Review: Google Maps Hacks

Wes Meltzer checked out the latest of O'Reilly's *Hacks* series. He notes that "one of the big points of the O'Reilly books has always been that they are very nearly as usable by the novice as by the expert. This book did not seem to be."

Review: XIII

Follow Agent XIII's pursuit of the president's assassin and his own past in this comic-styled first-person shooter. A conspiracy of 20 agents seeks to stage a coup and unseat the

US government, unless you can recover from your amnesia in time to stop them. David Duchovny and Adam West voice characters in this action mystery, adapted from a popular French comic series.

E-Mail

SmartBoard USB5000

Enjoyed your article on the SmartBoard. I have one and was vexed by the F12 key and its apparent inability to open the CD drawer, but upon talking with DataDesk they brought to my attention that the time the F12 was held down governed that. If you hold it down for a second, the drawer opens.

—Dave

Worms 3D

I love this game but absolutely *hate* the controls. It is the single most annoying problem that has kept me from playing this game. I have e-mailed the company begging for some sort of upgrade to allow custom control configurations, but their response was simply that most people have no problem with the controls. Maybe a group effort would convince them otherwise. I really *want* to enjoy this game, but can't due to the controls. I hope someone at Feral gets this message and helps me enjoy the game once again!

—Gary Katz

TVMini HD

I bought one of these and was really disappointed that the mini antenna included doesn't really work. You actually need to plug the device into your home roof antenna to get reception on all channels.

Unfortunately, because of this, it doesn't allow you to be mobile if you use it with your laptop. So, if you already have a TV at home, there's no point in getting one of these for your laptop or desktop.

—Jason Hartner

Axio Hardsleeve

If you orient the notebook with the latch facing the hinge of the hardsleeve, you wouldn't have the latch problem you described.

—Scott Jones

> You are, indeed, correct, however that would not be a workable solution for people who leave the laptop in the Hardsleeve while using the computer.
>
> *—Lee Bennett*

New ATPM Reader and Spoofed Mail

Just discovered your site today and read your current issue cover to cover. Can't imagine how it managed to stay "hidden" from me for all the years (nine) that I have been a Mac user.

What I would like to see in your online publication is an article on spoofed e-mail addresses, how this happens and what people can do (if anything) about it or to prevent it. I thought that Apple had security measures to prevent this from happening—there is one article on their support site about this—but I seem to be proof that it isn't so. This problem has just surfaced this week. I have already given up one e-mail address (not a .Mac though) because of this problem, and I don't particularly want to migrate to another. Any advice on this subject would be most welcome.

Keep going with a great online publication. I am planning to subscribe after I hit the "send" button on this e-mail!

—Kim

> Due to the way Internet e-mail works, the only way to prevent people from sending messages with your return address is to not let them know what your address is. So, it might help to use a separate address "in public" on the Web and when companies require it, reserving your main address for personal correspondence. You could file the messages for the public address separately and dump it if it starts collecting too much spam. However, I think the best solution is use a spam filter to get rid of the spoofed messages, along with the rest of your spam. My day job is developing SpamSieve, and that's the filter I'd recommend.
>
> *—Michael Tsai*

Cover

Hello Mirko! I love your ATPM cover for this month! Its delicate hues and soft "high-tech" look remind me at once of the "hall of mirrors" with its seemingly infinite perspective, and a favorite of faux-high-tech mid-eighties looks in interior design in the US; and your treatment also evokes for me the delicate "Shagreen," egg-shell/lacquer and "Coromandel" screens of the 1920s and 1930s designed by famous artists and designers such as Pierre Dinand etc. And the colors are just exquisite.

Thank you for such a lovely, retro/modern composition. It's one of my favorite covers!

—Catherine von Dennefeld

iDisk Ennui

Well-written! I am a recent "switcher" and I subscribed to .Mac in spite of some hesitation over its rather stiff price. I sought to experience the fully-integrated Mac world that my new machine that .Mac promised to offer. As my first Mac arrived with a sick logic board

and daily kernel crashes, Backup turned out to be a very useful utility for my many OS re-installs. I still find it a nice, automatic feature, doing daily what I would normally neglect to do. After my experience, the only reason I might recommend .Mac is to users who want to have one vendor for everything, and who find the poor potent for-pay Yahoo services too confusing. I do not know if I will continue my subscription at this time.

—Mike Reith

• • •

Thank you for a lovely commentary. (I even read the part about .Mac!)

I am a London-born Merseysider who has lived in California for many years. The ground is moving beneath my feet as this country seems to be changing for the worse—or I am just getting old.

To read of you sitting in your garden, listening to the distant bells, was too piquant for words. I am a semi-retired pastor and use .Mac and iWeb for a church Web site. I could not have created a site except for the ease-of-use of .Mac and iWeb.

The church pays for .Mac, and I save a little by buying our subscription from Amazon. I sincerely doubt I would buy .Mac for my own use. I do think the URLs on .Mac are clumsy and long, and people tell me the site loads slowly.

Thank you again for a most pleasant article. Please keep it up.

—Roy Gee

> Thank you for your fine comments. They were well received.
>
> I can assure you I really was sitting in the garden listening to the bell ringers and the bees humming. It was a quintessentially English experience—as was the wine, but that is another thing altogether.
>
> Today, I got an iCard from a Danish friend with links to iWeb photos of her rapidly growing family. She is just sharing her life with her contacts around the world, and it made me realize that there is a great value in the facilities offered by .Mac if one chooses to use them. Maybe I've been a little too cynical about .Mac.
>
> I'm just off back to my garden to watch tiny pipistrelle bats flit across the pond and fly between my wife and I as we talk. They fly so close and so fast it is magic.
>
> Or maybe it's the glass of wine I will be drinking... :-)
>
> *—Mark Tennent*

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.

Bloggable

by Wes Meltzer, wmeltzer@atpm.com

It's Time to Say Goodbye

Remember that kid you knew in high school, the one who started playing baseball or soccer or football when he was in elementary school and kept doing it until he was a starter on the varsity team? Or maybe he was a photographer, or a painter, or a writer, and just made it seem effortless. He was friendly, and he made a point of suggesting that everyone start doing it. Maybe you even took him up on the offer. Maybe you're still doing whatever it is that kid did.

Do you remember being surprised, a little startled, when you found out that he'd given it up—packed up his bags, and moved on?

Mark Pilgrim's highly publicized switch this month from the Macintosh platform to Ubuntu Linux on a Lenovo ThinkCentre made the kind of waves that only that kid from high school could.

Pilgrim was a longtime Mac power-user and hacker, and an early blogger, whose collection of software and writing was highly influential to a great many of us. Like me. I wouldn't say that I bought my first Mac because of Pilgrim's writing, but I think it's fair to say also that I might not have had the kernel of an idea that buying one was the right decision without his help.

So on June 2, when he announced that he'd walked into an Apple store and walked out without buying a new computer, and instead had bought a PC, many of us were genuinely stunned. It was not a hardware decision for him, but a software decision: as an open-source enthusiast, Pilgrim wanted open data formats and open code, and Apple has a history of providing neither. He wrote:

> And what about those wonderful Apple programs that I haven't replaced with open-source alternatives? I loved iPhoto until my iPhoto database got corrupted one day, and I lost all my ratings, keywords, and albums because that information is stored in an undocumented binary black hole. Yeah yeah, I know about AlbumData.xml. That has its own problems, and in my case it was already corrupted by the time iPhoto noticed. I'll give them some credit for trying.
>
> Similarly, I loved iTunes until my iTunes database got corrupted, too. Once again, I lost all my ratings and about two dozen well-thought-out interlocking "smart" playlists. And once again, all of the irreplaceable metadata was stored in an undocumented binary black hole. Yeah yeah, the XML backup again. iTunes even helpfully offered to restore from it... except that it didn't restore

> any of my aforementioned metadata, so it's not really a backup, is it? "A" for effort, "D-" for implementation.
>
> [. . .]
>
> I'm creating things now that I want to be able to read, hear, watch, search, and filter 50 years from now. Despite all their emphasis on content creators, Apple has made it clear that they do not share this goal. Openness is not a cargo cult. Some get it, some don't. Apple doesn't.

About as soon as he posted about this, the feedback started flooding in, many from Mac users who sounded angry and betrayed by the loss of a highly publicized figure. It's a little like finding out that your representative or senator, or the head of your political party, just switched to the other party. An aside: those of you who remember when Vermont Senator Jim Jeffords became an independent in early 2001 may recognize some of the screaming, angry vitriol in the comments on Pilgrim's initial entry.

That was where John Gruber got involved. He is, as always, an excellent and level-headed analyst, and his take on Pilgrim's argument can be reduced to a quick punch: he thinks it's worth exchanging a little openness for a better and more intuitive user interface, but not everyone does. Gruber makes the neat analogy that reminding the reverse-switcher that his new UI is not as good is "like telling someone who is switching from a Chevy Tahoe to a Toyota Prius that he's not going to have as much cargo room. He knows it."

Gruber seems to believe, as I always have, in the relativism of operating systems. Now, that's my facetious shorthand for saying that for each user there is a best operating system, and that it is not necessarily OS X. And if, like Mark Pilgrim, you are extremely worried about open formats and open source code, OS X cannot be the best operating system for you.

Mark Pilgrim responded with an exhaustive list of all the times that, in fact, his data or code *were* broken by Apple, starting in 1983. He also makes the fair point that extinct data formats are much more easily converted when they're documented.

But this wasn't just a two-person conversation, plus a bunch of jumping-up-and-down ranters in the comments box. It provoked the question, "Is it time for me to switch?" in Tim Bray and Ted Leung, who are both prominent Mac users and serious open-source evangelists as well. Bray breaks down his list into things that would already make it easy for him to switch, things that are sticky, and things that are keeping him on a Mac, notably projection and fast resume-from-sleep. I would note, though, that some of the best open-source software either runs on Macs or is OS X–native, like Adium and Firefox. He also thinks Apple should open-source some of its applications. (Gruber writes, later: Sure, but they won't.)

Leung suggests that it's just a matter of time before he starts using Linux as his primary operating system, that there's just a certain degree of spit and polish Linux needs before he'll port all of his data over. He, too, lists features that he expects from Linux: better font

support, scriptable applications, color management, something like iSync, and more great applications.

I'm not sure how possible these are. Eric S. Raymond's bazaar model may get the job done, but UI spit and polish is hard labor in the salt mines, and very few people will do that for free. On that note, Rui Carmo says that developing software is like cooking, where even if you have the instructions or source code, getting it just right requires a great deal of know-how, something the casual cook might not have. I think it's a reasonable point.

At any rate, good luck, gentlemen. And godspeed, Mark Pilgrim.

Bread Pudding

- What's up with the OS X kernel, and why hasn't the Intel version been made open source? There's been a lot of speculation, since Avie Tevanian left, that Apple is switching from Mach to something else, and that *that* is the hold-up. To this I say, bah, spare your oxygen. John Siracusa agrees: there are lots of good reasons (like important kernel changes for 10.5, or trimming out proprietary Rosetta code) for Apple to delay the release. But they *do* have to do it, sooner or later, to comply with the license. So keep your eyes open.

- The California Court of Appeals has ruled that California's shield-law privileges for journalists apply to the bloggers in the *Apple v. Does* case. *Ars Technica* has a fascinating analysis, and *Macworld* takes the court to task for being inconsistent with its application of the shield law. I don't know if I agree with *Macworld*, because the *spirit* of California's shield law is to protect reporters no matter what *kind* of media they work for, but it's still a persuasive argument.

- This month's nominees for the *JFK Shot by LBJ Award*: Toronto's *Globe and Mail*, for wondering whether Apple might partner with RIM, the maker of the Blackberry, to produce the "AppleBerry"; and *CNet's* Crave, for speculating that Apple might buy Nintendo. I am speechless. I have no speech. Moving right along.

- Would you believe that Dave Winer—yes, I know—caught John C. Dvorak on camera, saying that he deliberately tries to piss off Mac users in his columns, just to generate more readers? You must watch this video. It's just unbelievable. Gruber notes that he has long believed this was so, but that he was surprised to hear Dvorak admit it. The *Slashdot* thread, with its 272 mostly hilarious comments, is well worth a read.

- Windows Vista will now run on your MacBook Pro, too. Via a Microsoft staffer himself. Wow. One more way to get my whole family to switch. . .

- Another reason to love OS X: Sane application installers. Since I left the Linux world in 2002, I have very rarely had to ask myself, "Why must I install this dependency again?"

How do I love Call of Duty? Let me count the ways.

There's an ironic joke we make in England about public transportation. It's not about traveling on public transportation itself, which is, on the whole, remarkably good. In larger towns and cities, travel by bus is often the fastest and easiest way to get around because of the special clearways set aside for buses' and taxis' sole use. Our retired citizens also get free travel at off-peak times. The joke is about waiting for public transport. It always seems you wait ages for a bus to arrive, then two come along together.

As an ex-double-decker bus driver from when a career change 25 years ago saw me funding my way through college by driving buses, I understand the reason why they concertina together at busy times. It just doesn't help on a wintry evening in the rain when you are waiting and waiting and waiting. Now, however, this joke has worn a bit thin because modern bus stops have arrival times displayed on LCD screens, linked wirelessly from stop to stop. Potential passengers know to within a minute or so when their bus will arrive.

Two coming along together is exactly what happened the first time my brand new potato peeler was used. It has a razor-sharp stainless steel blade. When peeling the very first potato, it took the tip off one of my fingers. 15 gallons of blood later and swathed in plaster, meal preparations continued. Unfortunately, with one finger wrapped awkwardly and stuck out at an odd angle, the knife slipped and cut another finger. Not the first time with this particular blade either; it has tasted my blood on more than one occasion. I once saw bone where it had sliced me so deeply as I slashed at a thick broccoli stem in the vegetable garden. Luckily, it was not serious this time, though I'd prefer not to dwell on the subject or I might faint from the memory.

It also explains another unexpected pairing—my two typing fingers just happen to be the same ones that had suffered cuts and also the same two used in Call of Duty to turn right or fire a weapon. That made the arrival of Call of Duty 2 a little less of an event, especially as five minutes after the postman delivered the package, a courier arrived with QuarkXPress 7.

To say I'm a bit of a Call of Duty fan is to describe Australia as an island. It has been noted by others as well, as is shown in SuperDuper.

SuperDuper!

Heroic System Recovery For Mere Mortals.

Version 2.1.2 (v78.2)

Mands, Geoff BlackKeller, Kirk Van Druten, Nick Greeves and Scott Kramer.

Special thanks to our wives, families, friends and pets for patience, understanding and support during the long period of development, documentation and tweaking.

Extra special thanks from Dave to Bruce for putting up with tweak after tweak after tweak (after tweak)... and to Mark Tennent -- for service above and beyond the call of duty!

The last two lines say it all.

QuarkXPress has been a major part of designers' lives since 1989 and enabled a comfortable living for many. Only Quark's tardiness at getting a Mac OS X version drove the rise of InDesign 2, followed by CS1—both of which many designers enjoyed using—but perhaps not the latest CS2, which has proved problematic. As a consequence, QuarkXPress 7 has the hopes of many riding on it. This is, of course, another pairing that is not completely warranted—keeping updated copies of the two major industry standards such as XPress and InDesign just in case they're needed. At least Freehand and Illustrator have been able to open each other's files for years. If only Quark and Adobe would agree on this, too.

It's amazing to find that Call of Duty 2 and XPress 7 have a big thing in common. Both arrive with printed manuals, something of a rarity nowadays, and in Quark's case, extremely welcome. What a disappointment to discover that while one of the packages is better than expected, the other is a big letdown. Both have been covered elsewhere (see above) so this is not meant as a review, just notes from the first week of using them alongside each other.

First, the good news. QuarkXPress 7, while slated by many reviewers, seems an upgrade worth having. It is, if you like, the program that QuarkXPress 6 should have been. The redesigned tools and palettes make using it easier and more pleasant, especially for those with dodgy memories whose hard-learned XPress keyboard shortcuts have become blurred with InDesign's. New photographic controls make trips to Photoshop unnecessary for many effects, and XPress will open native Photoshop files as well. Any changes applied to images can also be saved as a workflow and/or only applied to the image at run-out time. Color control has had an upgrade, so it should mean the end of telephone calls from printers asking why the four-color job you sent has six additional spot colours because the logo supplied by someone else is not separating properly. The new soft proofing on-screen will also help this.

On the other hand, QuarkXPress 7 is still not fully Mac OS X–compatible so that Services and all the other useful little tools aren't available. The printed manual leaves out some topics altogether, for example, "check spelling." This one topic alone is a big weakness in XPress, in which spelling still has to be checked via a separate window, unlike true Mac OS X applications that have instant access to a dictionary, a thesaurus, and spell-check as you type. PDF creation is still via Jaws rather than a true Adobe engine, which probably means that XPress will export its own style of "nearly compatible" PostScript. There is, though, support for PDF/X-1a and PDF/X-3, which are must-haves for working in a modern PDF environment.

The Web site creation tools have improved. I have never built a site with XPress, though it has been possible since version 5, and even before using Myrmidon. To a great extent, QuarkXPress' Web tools have been surpassed by applications such as Freeway, which has commands similar to those in XPress to ease print designers into Web design.

Quark has also extended collaborative working tools, and this is where the basic version of XPress scores over InDesign. While both have special versions and tools geared for heavyweight users—such as newspapers and magazines—in their basic format, QuarkXPress is better for collaborative groups, while InDesign is more suited to individual designers working on their own. New Composition Zones return the control of page design to the designer. Now she can create a Composition Zone—such as a text box or story space—and send it to the editor to fill in. The editor will not misunderstand that "cut the length by two paragraphs" means the opposite and send 150 additional words instead. The editor will get a file showing the exact space his story has to fit into, complete with text and paragraph styles predefined.

Composition Zones are, in effect, miniature QuarkXPress documents that can be as simple as one text box or can be a whole page layout. One editor I work with insists on editing directly to the QuarkXPress document but has no understanding (or willingness to understand) style sheets, master pages, and the like. Now he can tweak the text to his heart's content without ruining the remainder of the document.

Call of Duty 2, sadly, is the big disappointment. It's like at Christmas when you asked for a Ferrari but end up with a Fiat. With Call of Duty 2, the graphics may be (slightly) better and gameplay a little different, but it is not a successor to Call of Duty United Offensive, the previous release in the series. The name itself gives the game away; it is an update to the original Call of Duty. For United Offensive players, gone are the broad arenas and burning rubber in a Jeep with a couple of tooled-up pals. There is a level with tanks (I haven't finished the whole game so there may be more), but it is very easy to complete, and that describes the game in general: it's a bit too easy and familiar. Each level is close combat and pretty much the same as the previous one. Grenades (yours and theirs) seem almost limitless and able to solve most scenarios that can't be completed by running around while blasting away with a machine gun.

The online play is currently much the same small townscape arenas with no interaction with vehicles and big weapons. They are set in desert or Europe, but apart from the color

of the buildings, land, and lighting, little else distinguishes one from another. In Call of Duty United Offensive, innovative designers have produced some evocative battlefields that even include working cable cars, submarines, and other features, as well as extending the weapons and vehicles. This is probably not going to happen in Call of Duty 2. Even the maps are similar to the ones in the original Call of Duty, and none seems able to conjure up the realism created by United Offensive.

The old saying states that the best things in life are free; in Call of Duty's case that ought to be One and Three but definitely not Two. Quark's new offering is another matter and a worthy upgrade from the late, great QuarkXPress 5...er...6.5. Be aware, though: old XTensions will not work with this new version.

Wherefore Art Now, Jonathan?

It may come as a surprise, but the first computers in the world came in three basic colors: pink, brown, and yellow, with a little variation in the shades depending on where they came from. At the start of the last century, real men did it with machines like the Burroughs. Although the machine was made of steel and tin and its operators grew one arm longer than the other from repetitive pulling on the lever, it was the men—not the machines—who were known as "computers."

The rise of the electronic computer such as Colossus, built by British telephone engineer Tommy Flowers, and funded mainly from his own pocket, saw the computer as a machine taking on the term as well as its operator. The visible glowing valves were a little taste of things to come. Even as recently as the 1960s, the Apollo space missions landed men on the moon using math worked out by rooms of pinkish, brownish, yellowish "computers" on their slide rules. These mechanical analog computers were invented nearly four hundred years earlier by Edmund Gunter of Oxford, England who devised a single logarithmic scale. His contemporary, William Oughtred put two of the Gunter scales together to make the first slide rule.

In the 1980s, Apple co-founder Steve Wozniak often credited with devising the world's first personal computer, besieged the world with beige. Then in 1987, Apple had one of those mind-boggling changes of heart and switched to computer cases made from the finest silver-grey. Meanwhile in the world of IBM and its Personal System/2, also released in 1987, beige was still the color of choice. This situation continued for many years, with computer manufacturers slowly joining Apple in the grey lobby.

As with all things, change had to come. Apple's computers had been through hard times, written off by many as too slow and unreliable as various machines came and went. Then along comes Steve Jobs and Jonathan Ive's candy colored Pop Art iMacs and all was well in the Mac universe again. Apple also changed the form factor of the cases, rounding off the bottom and rear encasing the cathode ray tube. Prior to this, most, if not all, monitors finished flat. Other Macs had followed more traditional computer shapes apart from the first Mac portable, a heavy and barely portable machine, and the unique design of the 20th Anniversary commemorative Mac. The latter was an all-in-one computer whose design wouldn't be seen again until the latest iMacs.

Jobs and Ive also introduced transparency. Our clients would stare at the Apple Studio Displays where all the workings were seen through the clear cases. They would crane their heads to see round the back as if something were hidden there and pondered on what the little lamps did. The cables also had transparent plastic sleeves. At the time, I wondered whether it

would be possible for computer cases to be transparent as well, and for little LED lamps to burn brighter as various areas of the computer did harder work. The cables could have pulsing LEDs along their length to show that data was passing through. Computing with Macs was for a time a little like Tommy Flowers' Colossus, with all the inner workings glowing.

In 2000 Apple introduced the much loved Cube. This innovative design was a forerunner of today's Mac mini. At Christmas 2000, Apple asked me to evangelize to prospective customers at a UK computer warehouse. The Mac area was streets ahead of the other PCs in the store and attracted a lot of attention. Children were drawn to iMovie and soon demonstrated Apple's famed ease of use. Given the choice, I think they would have bought Macs, but their parents would have none of it and purchased ugly and incredibly slow (by comparison) PCs. However, one thing that drew everyone's attention was the Cube. It was tiny and most thought it was the power supply for a computer hidden somewhere else. This may explain why it wasn't a big seller—people didn't understand it. Although I was not actually involved in selling the machines, it was incredibly frustrating for me when a wealthy worker from a North Sea oil rig came to buy a Cube and two Snow iMacs, then the top of range. The warehouse didn't have any in stock so I had to direct him to the Apple Approved Dealer opposite.

If Apple has been through beige, grey, candy colors, transparent, white and black, there doesn't seem much left for them to offer in terms of color, so will they have to innovate in form instead? To date, their Bauhausian designs have made form follow function—unlike their competitors who seem to do the reverse and bolt on extraneous lumps of multi-colored plastic in emulation of Apple's design, letting function follow form. There have been other attempts at look-alike computers following Apple's lead but none seems to have sold that well, apart from laptops which all follow Apple's designs, including moving to metal cases.

What next then? Palladian iPods? Cubist...er...cubes? Dadaist desktops? Gaudian G5s? Or even Gehry G5s, much admired by the likes of Brad Pitt, with scrunched-up and distorted shapes—that's Gehry's designs not Pitt.

Or more likely, back to beige.

About This Particular Outliner

by Ted Goranson, tgoranson@atpm.com

Outlining Interface Futures

This month's column deals with a rarely visited nature of outlining, focusing on some philosophy behind outlining and some user interface issues. We won't highlight a specific example application as we often do.

We still owe you the finish of the ConceptDraw Suite column, started last column. But alas, we were forbidden from using the insightful real examples we worked up, so there will be a delay while I make up an example to substitute. Meanwhile, a new version of ConceptDraw has been released.

The Motivation For Revisiting Structure

What we will tackle this month has to do with just what outlining can accomplish. It is prompted by several recent things that came my way.

The first one of these was BareBones Software's introduction of a new snippet manager, Yojimbo. You can read a good review of it in last month's ATPM. It competes with a few established applications that are tuned to snippet collection and management. Plus, there are quite a few of our power outliners that can incidentally do a good enough job of this for most folks.

Most of these employ the outlining paradigm, because, well, because it makes sense as a way of organizing huge amounts of information. In a way, these things do what the Finder would if it were better designed, so it isn't surprising that many use hierarchies with icons that look like folders.

Yojimbo collects all its stuff in a few "folders" that it calls collections. Hierarchy is not allowed. When it was first announced, the Yojimbo mailing list was ablaze with requests for nested collections (and other things). Bare Bones made a strong case for never having nested organization. (You can read the thread yourself.) Soon, enough influential users were hailing the "new" clean paradigm and eschewing outlines.

Now, these are smart guys that make good software. What they've done, I think, is hitch their wagon to a "new" notion of information tagging and retrieval that we'll see come into more visibility with the use of "extended attributes" in Leopard. Other file systems are beginning to use these, and the idea of having standard ontologies from the semantic Web means that you should be able to tag and find things easily. "Ontology" in this context means a standard vocabulary of terms and the relationships among them.

It's all connected to Steve Jobs' statement that the Finder is obsolete, meaning that hierarchical organization systems are obsolete. As with any such trend, advancing the new

unnecessarily means discarding the "old." Something to think about here at the *ATPO*, the church of hierarchies and structure.

Another event is the rather slow evolution of the outliners.org wiki, which is intended to serve as a community playground for those interested in these things: techniques, applications, futures. No, I have nothing to report on the progress of the wiki, as it is in the hands of a few busy readers. But I have been thinking about a section of the wiki that is dedicated to exploring and perhaps suggesting futures for outliners. There's a short list of features and directions that come up every time. But what about the cool new ideas that aren't so immediately obvious?

So I've been thinking about ontologies and structure. It's what I do for a living, too.

And the third event that motivates this column is Mori. Regular readers know that I shift around among power outliners. Call it a personality defect. Now it is Mori's turn to be heavily used at *ATPO* headquarters. Heavily.

Mori's developer, Hog Bay Software, has decided to go with a business model that is refreshing and a bit amazing. I bear no ill will to any developer, and all sorts are involved in the *ATPO* community. Every one I have encountered is worth knowing. Truly. But it is a bit maddening to be on the receiving end of a product development pipe and just getting what comes down, take it or leave it. In many cases, there is no warning at all that something is even in the works or that developer resources are applied.

The alternative seems to be open source, where if you so choose, you join a distributed team of mostly volunteers. You want a feature? Go out and program it. You think that conflicts with a competing philosophy of another contributor? Lobby your guts out on e-mail lists to try to convince others. I've tried it, and it frustrates. Questions of deep design often get watered down or avoided in open source.

I'd rather have a third way, where I pay someone to program and to bring some sort of coherent philosophy to the thing. But I get to advise. He consults me and takes me seriously. If something is hard or has a high cost, he lets me know and tells me what and why.

This is what Hog Bay is trying. It literally has a voting system where you enter your feature request, and the ones with the highest votes get implemented first. It goes so far as to report sales and Web site hits! This is more open than open software to me. Much of the source code *is* open, and Hog Bay encourages plug-ins. There have already been some impressive ones.

Well, you can imagine my dilemma. Here I am, a guy with ideas, and there sits someone asking. I've decided to put the more general of the notions from my ruminations here.

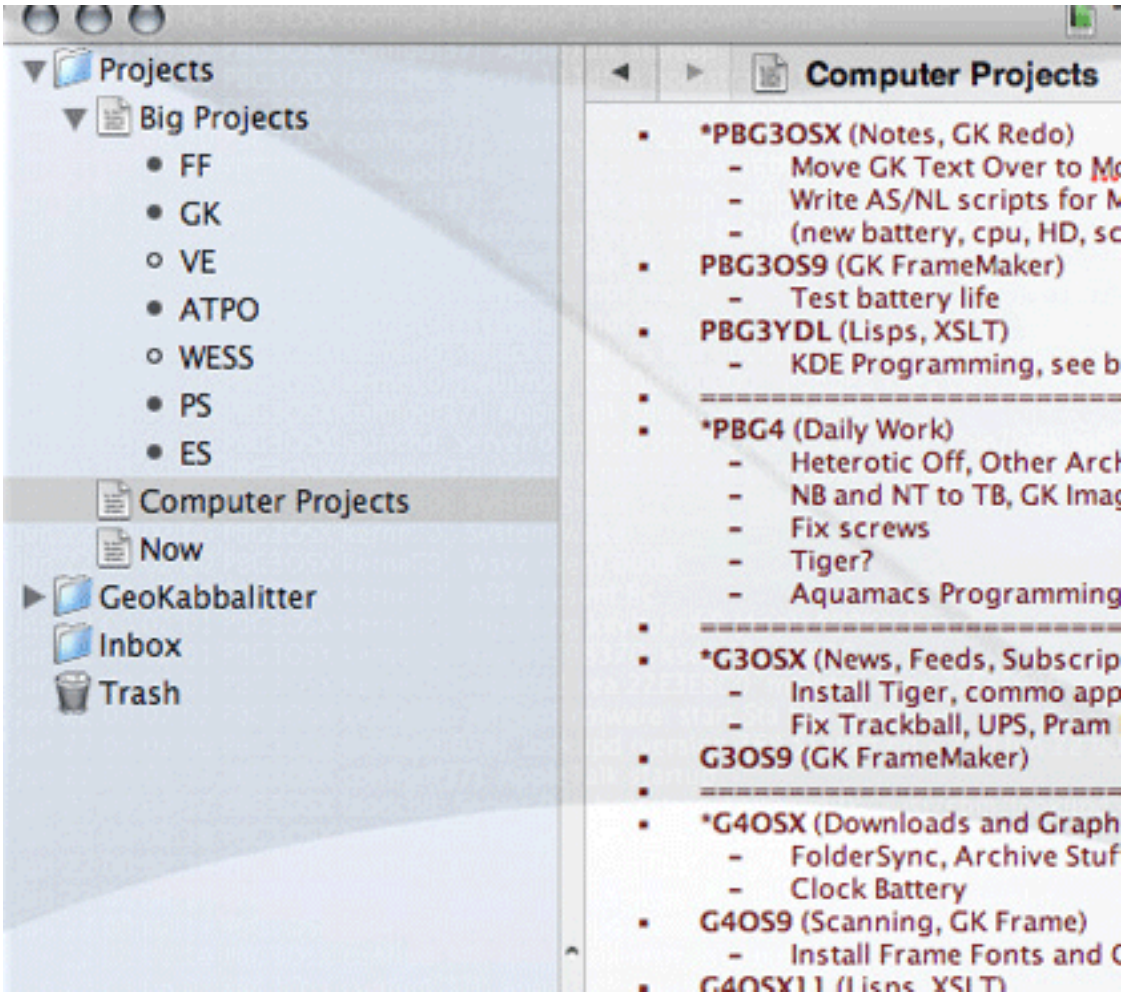
Assumptions

Outlining is familiar and natural. That's one of its attractions. But if you think of it, it is natural because it is familiar. It is so familiar in fact that we sort of constrain what we expect.

Let's go back a bit. Apple popularized the desktop metaphor, which Microsoft and now everyone else copied. That metaphor is pretty simple: you have a high level space "on" which you have folders. Folders contain other folders and files. That's the rudiment. The metaphor is simple, the notion of "in." Things are "in" other things.

Real files on the disk aren't in anything of course; this is just a way of organizing them. Now along comes the Finder's outline view so you can see what is in what.

Paradigm number one: outlining shows what's *in* what. That's the first paradigm which leads to the first limiting expectation, that being "in" something is like physical enclosure.

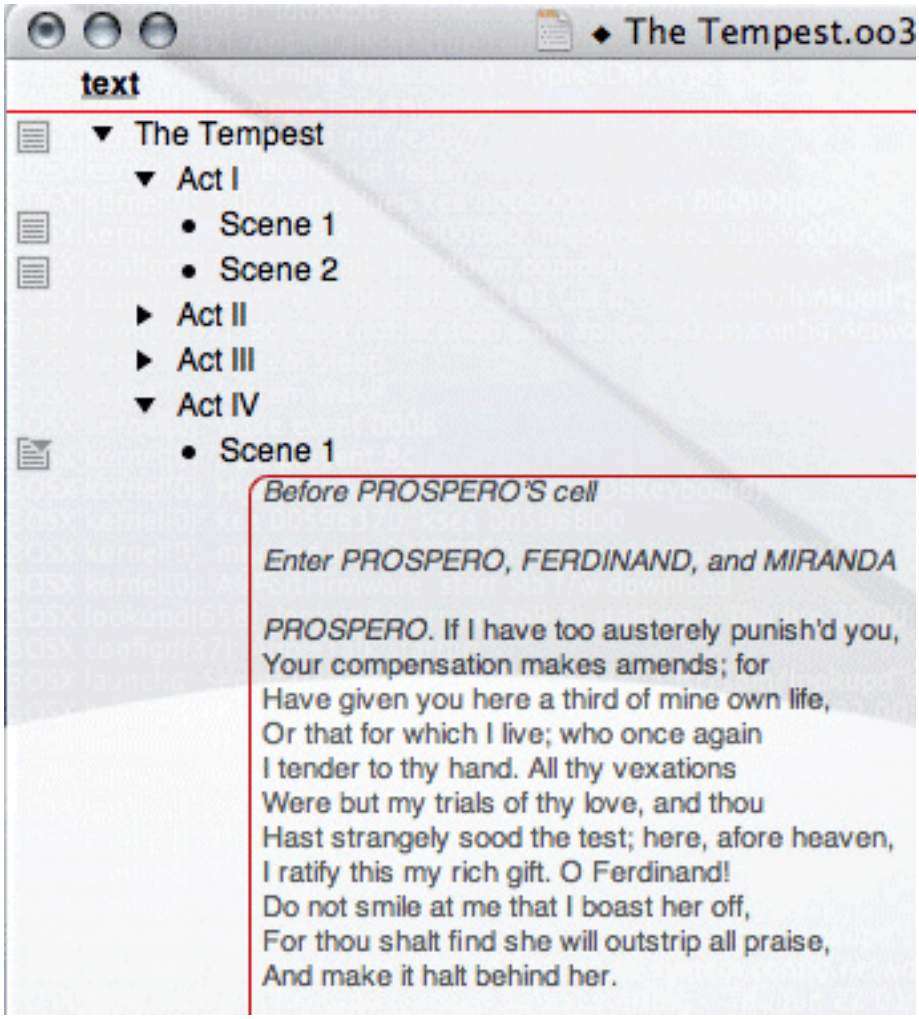


Files and Documents

Our second of three existing paradigms is the document paradigm. We wrote about the history of this in an early column. It comes from structured documents where you have chapters and sections, and perhaps subsections then paragraphs. Some documents, like technical and military manuals, are heavily structured for obvious reasons including introducing change. An outline provides a high-level view of the document structure. So, for

instance, you know in your car manual which chapter deals with the engine, and within that the fuel system, and so on down to the specific item of interest to you.

OK, so outline paradigm number two is that the outline shows finer detail. Or rather, the outline is a way of creating document structure, but in the user interface may function a lot like the paradigm above, where a desired paragraph can be found by looking "in" such and such section and subsection.

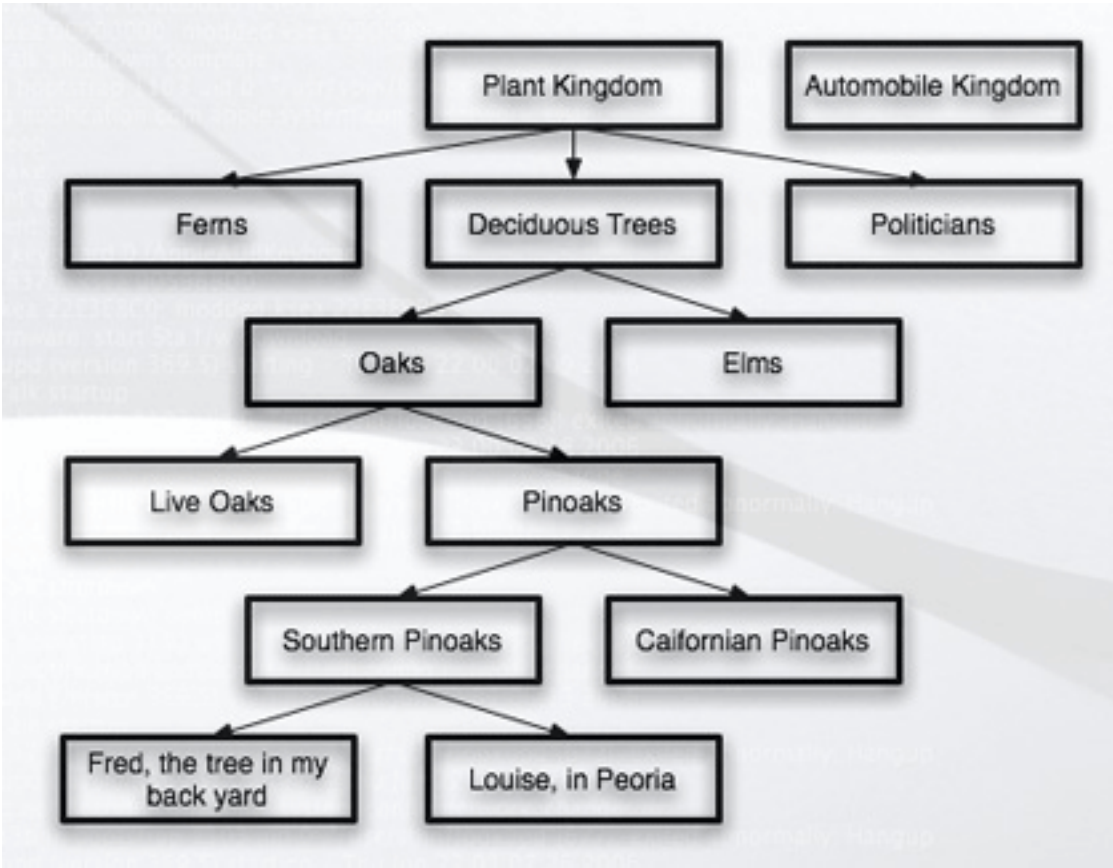


Structured Documents

The third could be seen as a synthesis of the other two, but I prefer it otherwise. This comes from the database world. Its the old Aristotelian model of classification: plants, trees, deciduous trees, oaks, pinoaks, southern pinoaks, that southern pinoak in my back yard. This is a matter of what something is and how to define it. It's different from the others, usually.

Outline paradigm number three is classification hierarchy. I won't say much about this because it is fraught with technical arguments. Let's just say that no matter what the

limits and philosophical problems, it's almost always the scheme of first resort when people organize things: lists and nests.



Tree-structured Data

Power outliners use all three of these, often with the same product but rarely for the same task or project. All three have a long legacy that goes way back before computers. The bad news is that these are so deeply rooted that we have a hard time escaping the assumptions that are behind them. Unless we have some clever thinking, we may be locked in these ruts forever. The longer we go without expanding, the deeper those ruts become and the harder to escape.

The interesting thing is that so many new products are appearing. They are amazingly varied and innovative, but not in the way they use outlining. No one anywhere is escaping the limits or expectations imposed by these three legacies.

Well, as the users that should be driving this community, we'll fix that, right?

Other Organizational Tools

What else is there? Lets break it down into two categories because outlining lives in two worlds. One world is the logical world. If something is the child of something else, that

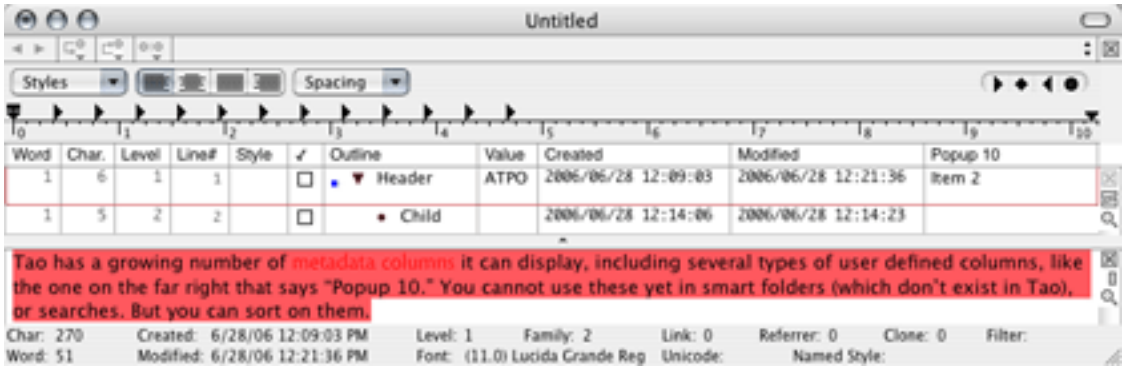
childness is a relation or attribute. But outlining is one of the few assignment conventions that also lives in the user interface world, where you can "see" the structure or relationship and browse in some way.

Logical tools can easily be described by reference to language, because in a way language is the representation we most commonly use when working with meaning. So, roughly, if you can identify something, describe its characteristics, its state (a complex one, that one), and how it relates to or changes other things, there'll be a common logical convention that can readily be used.

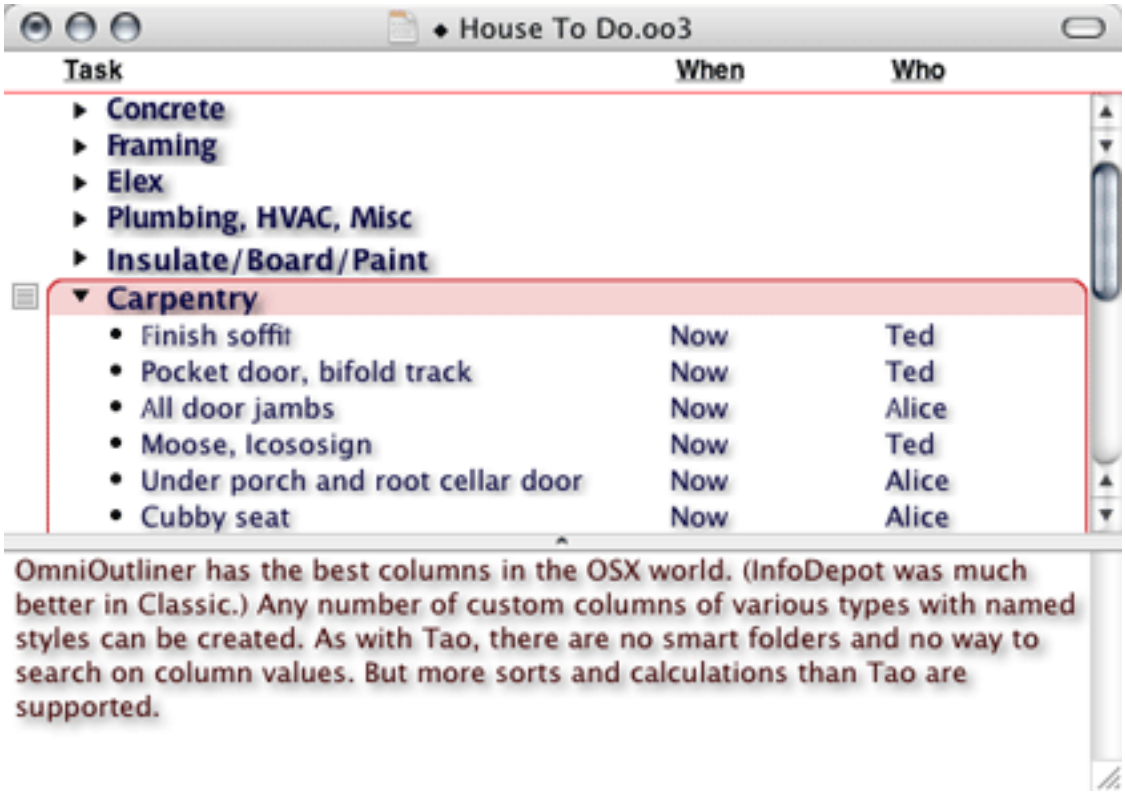
It's the user interface equivalents that are more limiting. We have a very few ways of displaying information. Let's count written language in here and equations. Both are fairly graphical, using visual means of different sorts. But they're pre-computer.

What we have is tables (including lists and columns), labels, annotative tags, styles, networks, and diagrams.

OmniOutliner, TAO, and Mori employ the user-defined columns paradigm, though not as thoroughly, to appear as an actual table or spreadsheet.



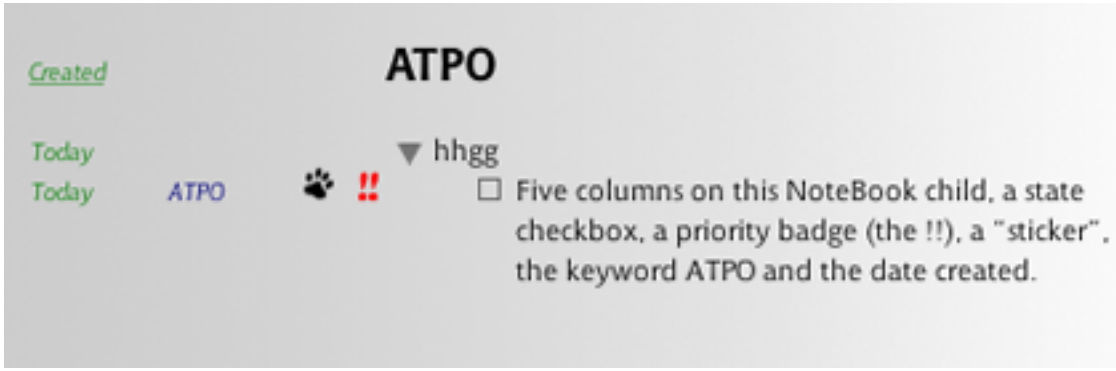
Tao's Columns



OmniOutliner's Columns

Almost all the products uses labels, following the notion of the Finder that a visual object has a color assigned. Sometimes these can be more than a short list and have user-specified names and colors. But only one such label per item is allowed. We'd need a new manner of visually displaying for more than one.

But labels can be seen as one graphical convention for a larger class of annotative tags. Products struggle with this, attaching keywords, categories, checkboxes, icons, and badges. Sometimes these have their own columns. All of these are logically the same.



"Tag" Columns in NoteBook

None of the products uses styles as metadata except a few that allow text highlighting to be a searchable and somewhat user-definable attribute.

Networks and diagrams are a tough one. There are a number of formally-defined modeling methods, and these can capture meaning well—many of them—to do useful work. But for general purpose use we end up with mindmaps and free-form diagrams.

Yojimbo and Leopard's Finder will probably use labels and columns.

The bottom line is that we have tons of logical tools to annotate and characterize things and few graphical conventions to display them in a way other than simply: colors, icons, keywords.

Oh. And then there's outlining.

Swallowing the Whale?

I put you through that survey—somewhat boring, I know—so that I could suggest that we start thinking about a next generation of outliners, ones that perhaps escape the restrictive usage we currently have. The downside is that the natural familiarity will be lost for some time while users accept the broader reach.

Here's the simplest idea, one I'd like to see immediately. Smart folders are a good idea. They extend the outlining idea in pretty radical ways. Few outliners at present have them—Mori, Tinderbox, Dossier—but I expect many to do so soon. The ones that do collect notes do so without regard to the hierarchy in which they rest.

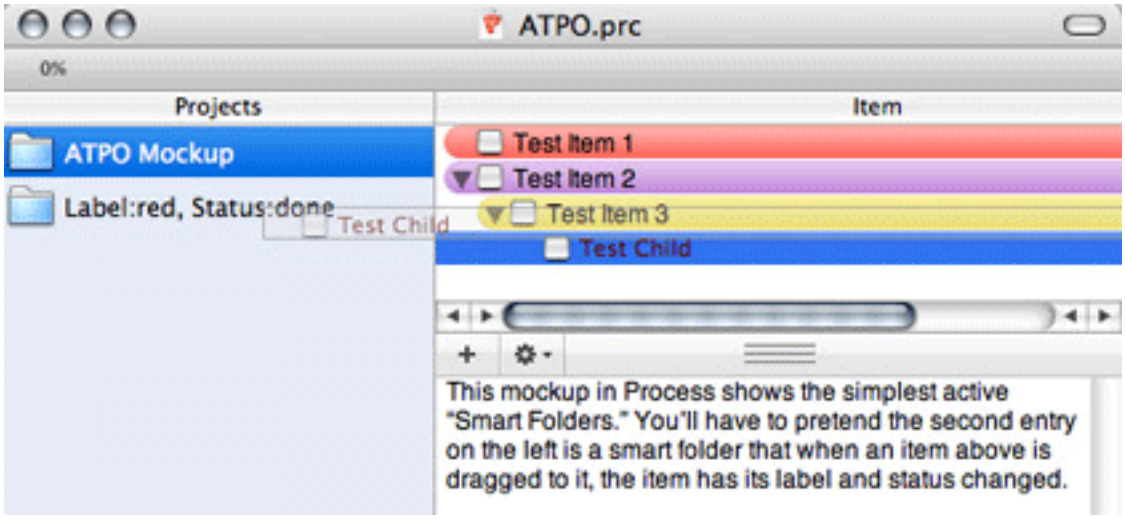
Clones are a related matter. Tinderbox only allows you to clone a note and place it somewhere. Its children don't go along with it. Mori's clones do carry their children, but clones and what ends up in smart folders are two different things.

What I'd like to see is smart folders that can find any combination of things based on any logical criteria and any metadata. We'll leave out for now two interesting elements of this: alternative logics and how notes might inherit metadata by links to objects outside the outline. For now, let's suppose that we really can have smart folders that find notes, make clones, and carry those clones and all the children of the clones. (Obviously, you'd want to sort out double entries where both a parent and child are found.)

No product does this now. Set it aside for a moment.

Now let's suppose that instead of the smart folder being passive, it were active. Here's an example: search for all the notes with a red label. Now you have a smart folder with clones of all the notes. That's passive. If it were active, I could drag a note to it that had an orange label and its label would change to red and an alias be added to the smart folder.

If I had two smart folders, one each for orange and red labeled stuff, then dragging an aliased note from one to the other would have a predictable effect.



Mockup of Simple Active Headers

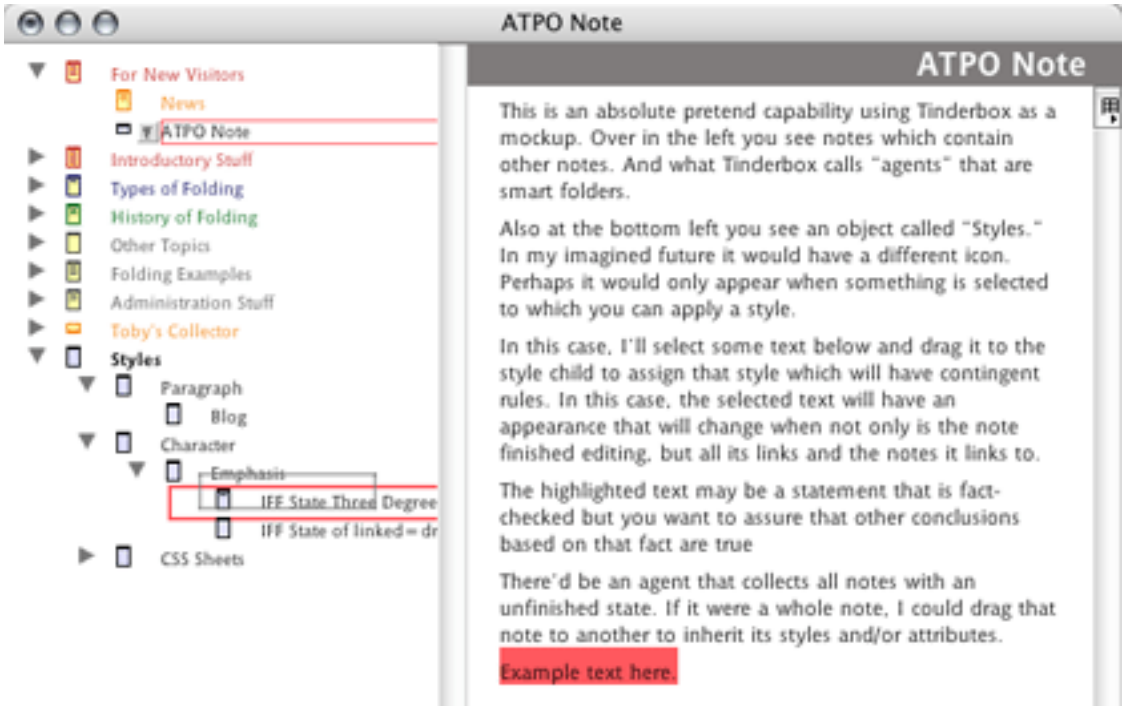
Now suppose that we weren't dealing with simple labels, but with more complex attributes. The outline would be a way of creating content, dealing with structure of that content, defining metadata for that content, and displaying and assigning that metadata. All this.

Suppose you wanted to escape the limits of the simple attributes that Leopard will use (and that are already more capable than anything used in any *ATPO* power outliner), and you wanted to go with words that changed meaning depending on context.

Now I ask you to recall smart folders with hierarchy.

A trivial example might be in building a task list and assigning an attribute to to a task to trash the note when certain conditions are met. So it is tagged conditionally. If it finds itself collected by an agent that satisfies those conditions (like all the dependent and linked notes are ready for publication), then its state changes. Tinderbox can do this now in simple cases. But what if it's not the state of linked notes you need to track, but the larger context of some derivative in the outline?

You'd need an inheritance hierarchy to show that, right?



Mockup of Active Headers

Now we are getting to the point that instead of simple keywords we can have conditional states and complex semantics assigned to a note, or a family or any group of notes.

Hey, suppose you want to work inside the note and assign attributes to a text block like we suggested in a prior column? Cascading styles? Defined in an outline. Assigning them? As simple as dragging a text block to the stylesheet parent and dropping it on the appropriate child.

Scripts? The same. In fact, combining styles and style sheets in this way is a natural. Whether we like it or not, XML is what we need for portability and longevity. And XML, my friends, is a specification for nesting assignments, like the ones we've been talking about.

Maybe with all the new types of outline objects, you'd want columns of outlines with some indicator of function, like we impishly suggested in a fictitious April Fools column.

And that's stuff that is probably doable now. Now. What you lose is the absolutely foolproof simplicity that outliners have now. That's not to be tossed away lightly. But what you get is something that could be useful to Yojimbo, or Mori, or its inevitable competitors, and probably something worthy of the outliners.org wiki page on suggestions for futures.

Leo, which isn't very friendly to Mac users, can have aliases that carry children, and incidentally can have themselves as children. And any note can have Python code that executes, modifying the behavior of the outliner. It's the only non-Mac outliner I know that can do something not yet found on the Mac.

Links

OK. So we've talked about two kinds of enrichment: logical things that can be said about other things, and an expanded sort of outlining structure of the second sort of those things that includes the first sort.

There's a third method of enriching, structuring: links. Several outliners allow links from text blocks to notes, more or less in emulation of Web links. Some others allow links from text blocks or notes to text blocks or notes. Tinderbox allows the links to be "typed," which in this context means named or labelled.

Even the simplest notion of links, in the right hands, can add a huge amount of structure. One might say that the parent-child relationship is a specific type of note-to-note link. Going further, if you just focused on text-block-to-note or note-to-note links, you could easily display and assign them using the drag to active smart folder strategy above. As new levels of the hierarchy sprung open, you could go arbitrarily deep in the link network, even setting rules like highlighting the links that follow higher ranked notes according to a search strategy, or even a Bayesian learning agent.

My point in all this:

If we relax our presumptions a bit, and see outlining as a powerful user interface convention, it can be used to satisfy all sorts of tagging, structuring, and enrichment assignment and browsing. And perhaps for many purposes and users, it may be a preferred choice.

There's a lot of life in the outliner world yet, and we are just at the beginning of a new era in how we interrelate and comprehend our information.

The ATPO Tracker

MindBurn

All hail the Mac! It's a constant theme of *ATPO* that the Mac is the place for outliner experiments because more clever and adventurous users seem to gravitate to the platform. So we can expect lots and lots of different twists on the outlining idea. Here's one.

What we have is an outliner that seems like many others in how it handles notes. As an outliner, it is a modern one, using Core Data and many of the same user interface conventions as Mori. But as an outliner it isn't in the power outlining class. Its claim for your attention is the ability to expose you to entries over and over according to a schedule.

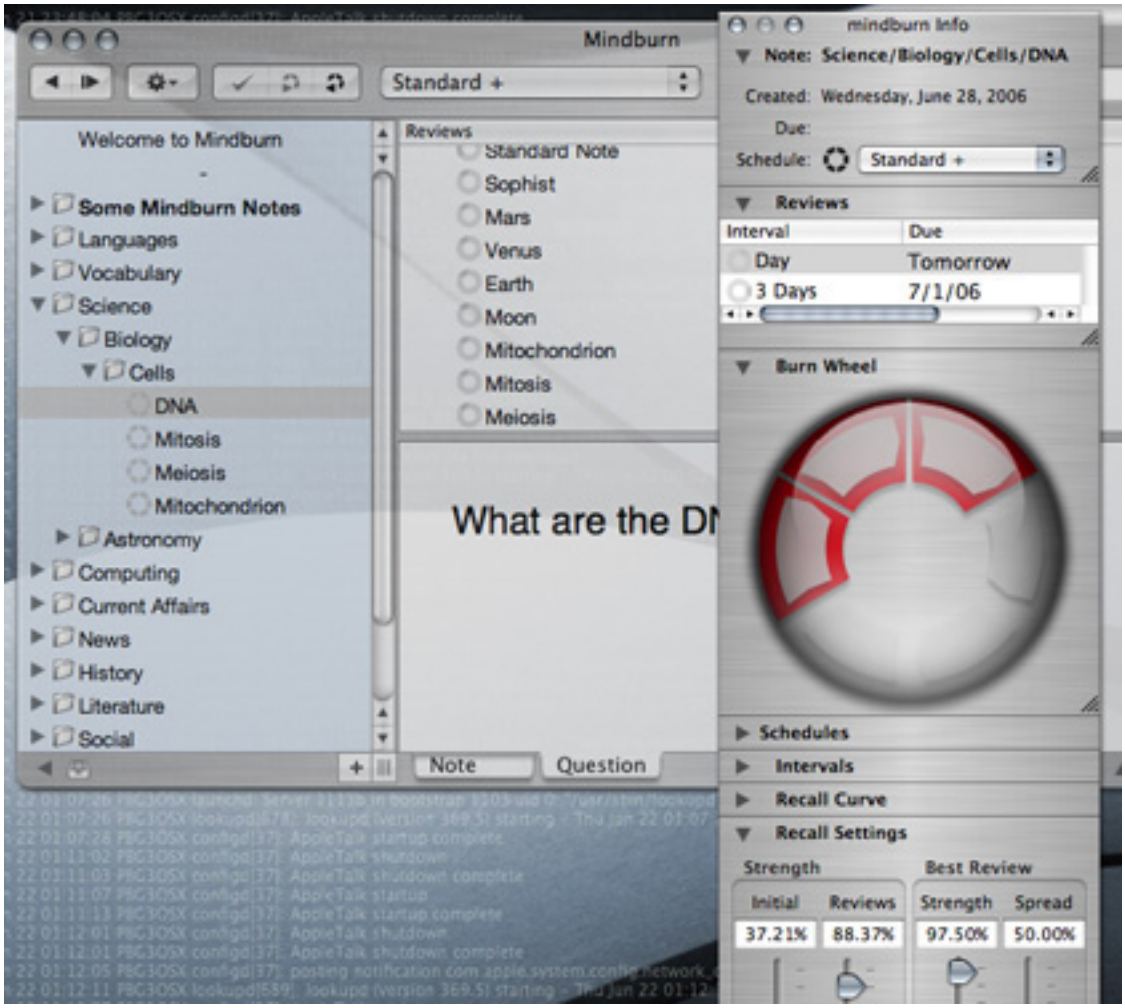
The theory is that any note worth saving is worth remembering. Another theory is that repeated exposure to a note "burns" it into the memory. Both of these are provocative notions. I don't believe them myself, but I'm probably in a minority based on the number of kids I see highlighting nearly all their textbooks. If you are inclined to believe these theories, take a look.

There's a separate issue here that MindBurn raises. Picking an outline tool involves deciding what you think of your information and where you draw the boundaries of useful information packets.

Some of our power outliners think of notes as if they were discrete entities, files in a way. MindBurn surely does. DEVONthink is in this school as well, and in fact can share resources with the Finder.

Others consider the whole outline as the thing that has identity. Writer's tools like Inspiration, TAO, or OmniOutliner (when used in this mode) are like this. It's the whole thing, each note in context of the others and the structure of the elements, that matters. "Burning" one of the parts of this misses the point that it is part of a greater whole, and that whole might be what needs to be exposed.

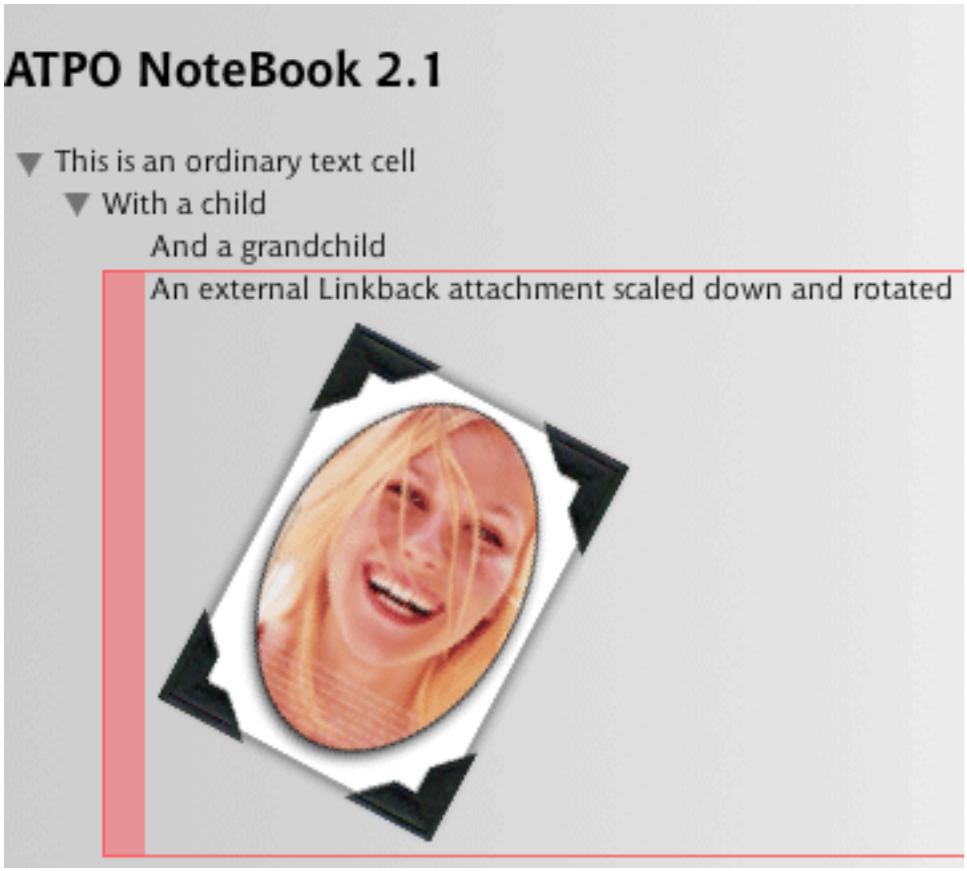
Others like Mori, Tinderbox, NoteBook, or NoteTaker can go either way, though Tinderbox has a unique philosophy in this regard. Some of these products are more flexible in this matter than others, and the latitude on drawing fences around information elements is something you'll want to consider.



MindBurn

Circus Ponies NoteBook

The by now venerable NoteBook has had a point upgrade to 2.1. It adds LinkBack support, discontiguous highlighting, and major improvements in its media frame support. Media frames are unique to NoteBook, I think. Now you can frame more things and better.



NoteBook's Media Frames

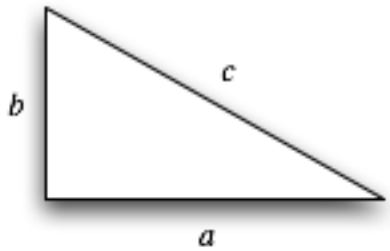
by Charles Ross, cross@atpm.com

Script Parameters and Results

Welcome back to *FileMaking*. This month, we're going to continue our investigations of FileMaker's scripting capabilities that we began last month. Now we'll take a look at two features of ScriptMaker that were introduced in recent versions of FileMaker.

Most programming languages have a construct that is very similar to FileMaker scripts. These are known as subroutines, methods, or handlers, depending on the language one is programming in. Regardless of the name, most languages have a feature that comes with these subroutines: information can be passed to the subroutine when it's called and information can be retrieved from the subroutine (in which case it's often then called a function). When information is given to a subroutine, the information is called a parameter or argument. The information returned by a function is the result. Parameters and results make subroutines much more powerful. Rather than simply executing code, the routine can make complex decisions and behave very differently depending on the parameters passed, and once the subroutine is finished executing, it can communicate back to the calling program what it has accomplished.

A simple example: suppose that while writing an AppleScript program, you wish to calculate the length of the hypotenuse of a right triangle. To refresh your high school trigonometry, the hypotenuse is the longest side of a right triangle, and its length can be calculated by squaring the lengths of the other two sides, adding these two squares together and taking the square root of the sum.



$$c = \sqrt{a^2 + b^2}$$

One could write a simple AppleScript handler that is passed the lengths of the two sides as parameters and returns the length of the hypotenuse (note that raising a number to the power of `0.5` is equivalent to taking that number's square root).

```
on HypotenuseLength(a, b)
    return ((a ^ 2) + (b ^ 2)) ^ 0.5
end HypotenuseLength
```

Such a function could then be called elsewhere in the program with something like this:

```
set hypt to HypotenuseLength(3, 4)
```

After being called, `hypt` would have a value of `5` ($3\times3 + 4\times4 = 9 + 16 = 25$, the square root of which is 5).

Until FileMaker 8, this kind of functionality, while possible, was difficult at best. Translating the above function to FileMaker 6 would require three global fields: one for each of the parameters and one for the result, and would look something like this:

```
Set Field[gResult, ((gParam1 ^ 2) + (gParam2 ^ 2)) ^ 0.5]
```

If another script wished to make use of this "function," it would do so like this:

```
SetField[gParam1, 3]
SetField[gParam2, 4]
Perform Script[Hypotenuse Length]
SetField[Length, gResult]
```

As you can see, while you can do it, generally you don't, because it has so much overhead to get parameters into Pre-7 FileMaker scripts and get results out.

Script parameters were introduced with FileMaker 7, and script results with FileMaker 8, allowing us to now create (almost) honest-to-goodness script functions in FileMaker using ScriptMaker. I say "almost" because, as yet, you can't simply assign the results of a script by calling it, as you can in a more traditional programming language, and strictly speaking, FileMaker scripts can only take a single parameter.

The single parameter limitation is resolved with the use of custom functions. If you'll recall from a previous column, FileMaker Pro 8 Advanced allows the creation of custom functions that, from within a script, operate just like FileMaker's built-in functions. By defining a format for a string to contain multiple parameters, we can use a custom function to extract those multiple parameters from within a script.

There are many solutions to this, but here is what I use: whenever I need to pass a parameter to a script (even if it's only a single parameter), I use the format `"Parameter1 = \"Param1\"; Parameter2 = \"Param2\""`. Remember that a string is defined in FileMaker as a set of characters between double quotes. The `\"` pair of characters indicates a

single double-quote character within a FileMaker string, rather than closing the string as the last double-quote does. This is a common technique in programming called "escaping," where we are using the backslash to escape the normal meaning of the double-quote. Therefore the above FileMaker string has a value of `Parameter1 = "Param1"; Parameter2 = "Param2"`.

What you may notice is that this format could be evaluated as the setting of variables in a `Let()` function, which is exactly how we're going to use it.

We'll define a new custom function called `GetParameter()` which will take a single parameter, `ParameterName`. It's defined as follows:

```
Evaluate( "Let ( [ " & Get( ScriptParameter ) & " ]; " & ParameterName & " )"
```

This is a rather advanced use of FileMaker functions to create a custom function, so let's go through it slowly. First of all, let's assume that we have a FileMaker script called `HypotenuseLength` that takes a script parameter. And assume that we have called that script and passed it a parameter of `"a = \"3\"; b = \"4\""`. For the moment, know that if we call `Get( ScriptParameter )` from within that script, we will be returned that string. Given all of that, let's see what the `GetParameter()` function will do with it if passed the parameter `"a"`.

We have two substitutions to make to figure out the result of our custom function. The first is the part that says `Get( ScriptParameter )` and the second is `ParameterName`. Since we passed `"a"` to the custom function, substituting it for `ParameterName` results in:

```
Evaluate( "Let ( [ " & Get( ScriptParameter ) & " ]; " & "a" & " )" )
```

Not too difficult. Now let's substitute the value of `Get( ScriptParameter )`. Remember that this function returns the parameter pass to the script it is called within, so `Get( ScriptParameter )` will return `"a = \"3\"; b = \"4\""`:

```
Evaluate( "Let ( [ " & "a = \"3\"; b = \"4\"" & " ]; " & "a" & " )" )
```

We can see from that (if you look at it carefully) that we have one long string (returned from the concatenation of a number of strings) which is being passed to the `Evaluate()` function. Let's concatenate the strings to simplify the expression.

```
Evaluate( "Let ( [ a = \"3\"; b = \"4\" ]; a )" )
```

So, the string we are passing (after taking into account the escaped double-quotes) is `Let ( [ a = "3"; b = "4" ]; a )`. This is a valid FileMaker expression in itself. What we have done is used FileMaker functions to build a FileMaker function. We then use the `Evaluate()` function to, well, evaluate this expression. Our original expression becomes:

```
Let ( [ a = "3"; b = "4" ]; a )
```

The expression uses the `Let()` function to set two variables, `a` and `b`. It then returns the value of the variable `a`, which in this case is `"3"`, which is the final result of our original expression.

When we do this manually, it takes quite a bit of thinking to figure out what is going to happen. The fortunate thing is that we don't have to do this ever again. The function works, and if we pass a parameter to a script in the manner described, our `GetParameter()` custom function can always extract whichever parameter we want.

Now that we have the means to pass and extract multiple parameters in scripts, we can rewrite our FileMaker script `HypotenuseLength` as follows:

```
Exit[((GetParameter("a") ^ 2) + (GetParameter("b") ^ 2)) ^ 0.5]
```

The FileMaker script step `Exit` exits the execution of the current script and returns control to the calling script. It has an optional parameter that, if specified, the results will be retrievable with the `Get( ScriptResult )` function. While the above refinement doesn't simplify our original script very much (it's still a single line of code), the calling and retrieving of the script is much simpler.

```
Perform Script[Sub-scripts, HypotenuseLength; Parameter: "a = \"3\"; b = \"4\"
Set Field[ Length, Get( ScriptResult )]
```

You might be thinking that, after all this discussion, the task we've performed would be better done with a custom function itself, and you're absolutely right. I've used a very simplistic example to demonstrate the concepts. In contrast, let me give you a real-world example.

In building FileMaker solutions, I often provide buttons that let the user navigate to the first, previous, next, and last record in a found set. Before FileMaker 7, this entailed not only four buttons, but also four scripts, one for each of the desired destination records. These scripts were very simple, and, aside from some standard steps that I tend to include in all of my scripts, were a single script step, such as `Go to Record/Request/Page [ First ]` for navigating to the first record in the found set.

Scripts multiply easily in complex FileMaker solutions. I've worked with solutions that had hundreds of scripts. Any time two scripts can be combined because of similarities, one should probably do so, and the four scripts needed in pre–FileMaker 7 are all very similar. They all go to a record using the `Go to Record/Request/Page` script step, only the parameter they pass to that script step differs. Therefore, in my more recent solutions, I've combined those four scripts into a single script, called `Go to Record( WhichRecord )`. This new script takes a single parameter that has one of four values, and looks like this:

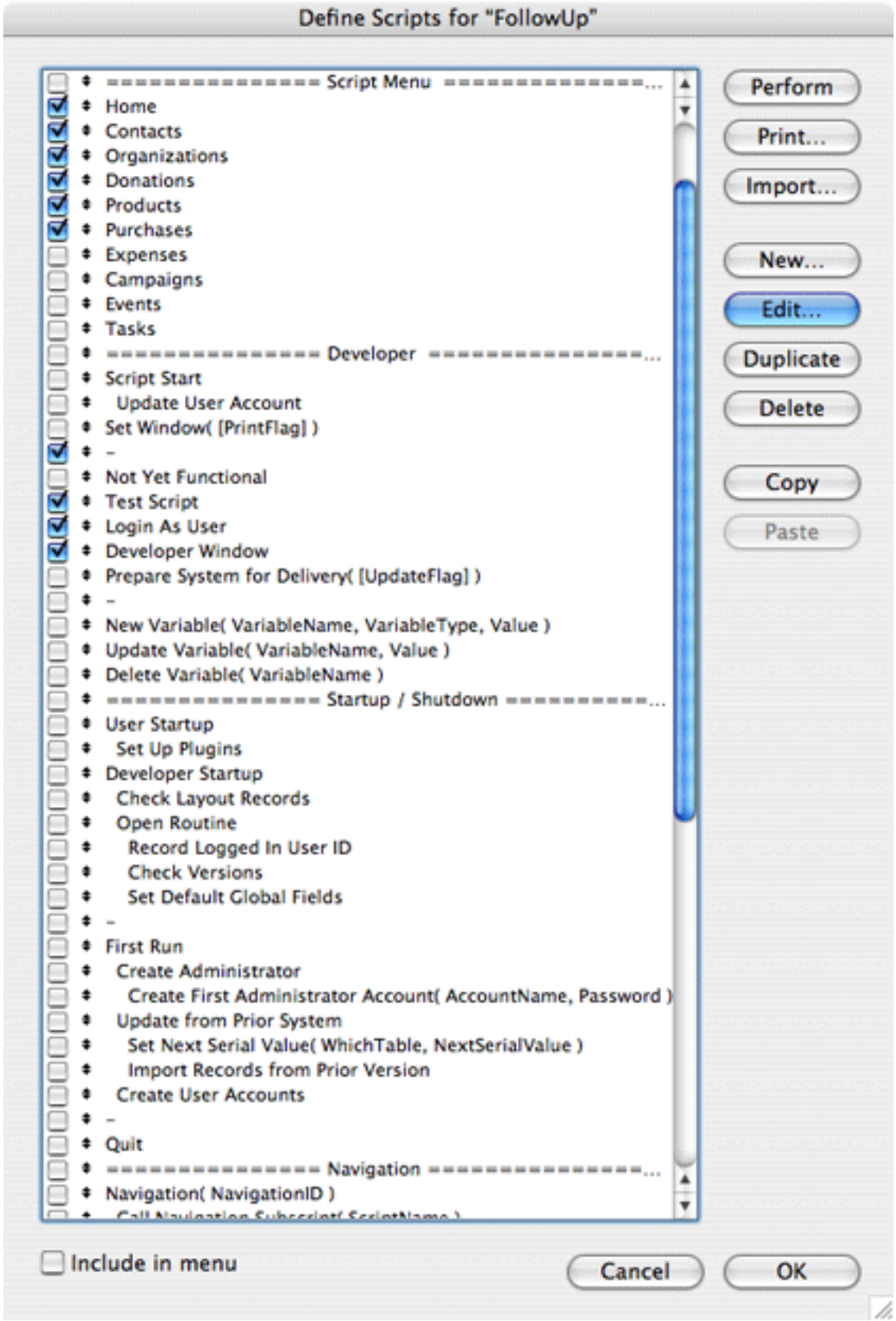
```
If[ GetParameter( "WhichRecord" ) = "First" ]
  Go to Record/Request/Page[ First ]
Else If[ GetParameter( "WhichRecord" ) = "Previous" ]
  Go to Record/Request/Page[ Previous ]
Else If[ GetParameter( "WhichRecord" ) = "Next" ]
  Go to Record/Request/Page[ Next ]
Else If[ GetParameter( "WhichRecord" ) = "Last" ]
  Go to Record/Request/Page[ Last ]
End If
```

When I want to call this script, I attach a button to it and have the button pass the appropriate parameter, such as `"WhichRecord = \"Previous\""`.

Scripting Tips

So far we've covered some very heavy material in this column, so I'm going to give you some time to absorb this and post questions if I haven't been clear in any way. But before I leave you, let me share some scripting tips that I've picked up over the last decade of working with FileMaker.

My first tip is: organize your scripts. Even with the script consolidation capabilities that script parameters provide, complex FileMaker solutions will still have dozens of scripts. I organize my scripts with two techniques. First of all, I create dummy scripts that have no script steps to act as headings and dividers in the "Define Scripts" dialog box.


Define Scripts for "FollowUp"
========== Script Menu ==========...
Home
Contacts
Organizations
Donations
Products
Purchases
Expenses
Campaigns
Events
Tasks
========== Developer ==========...
Script Start
Update User Account
Set Window([PrintFlag])
-
Not Yet Functional
Test Script
Login As User
Developer Window
Prepare System for Delivery([UpdateFlag])
-
New Variable(VariableName, VariableType, Value)
Update Variable(VariableName, Value)
Delete Variable(VariableName)
========== Startup / Shutdown ==========...
User Startup
Set Up Plugins
Developer Startup
Check Layout Records
Open Routine
Record Logged In User ID
Check Versions
Set Default Global Fields
-
First Run
Create Administrator
Create First Administrator Account(AccountName, Password)
Update from Prior System
Set Next Serial Value(WhichTable, NextSerialValue)
Import Records from Prior Version
Create User Accounts
-
Quit
========== Navigation ==========...
Navigation(NavigationID)
Perform
Print...
Import...
New...
Edit...
Duplicate
Delete
Copy
Paste
Include in menu
Cancel
OK

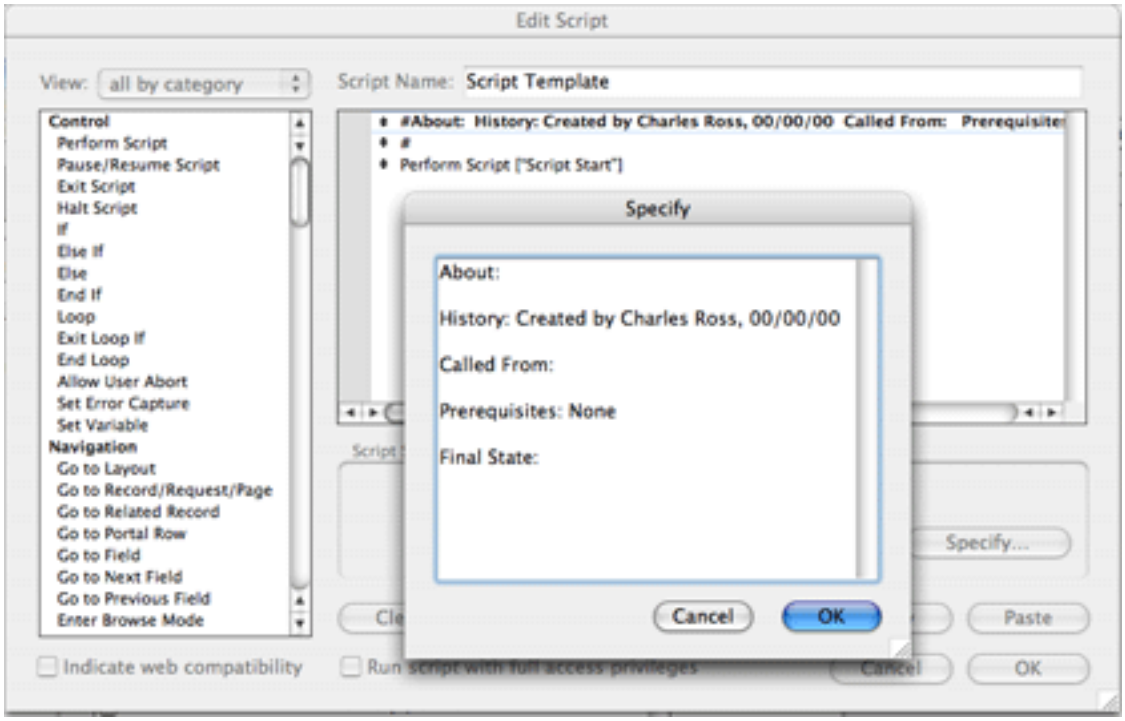
Every script that begins with an equal sign or is simply a dash is empty and is not meant to be run. It exists solely to provide structure and organization to the "Define Scripts" dialog box. Every script falls into one of the following sections: Script Menu (for those scripts that are to appear in FileMaker's "Scripts" menu), Developer (scripts that are useful to me but not necessarily to end users), Startup / Shutdown (scripts that execute automatically when the file is launched and closed), Navigation (scripts that take the user from one layout to another), Operations (scripts that perform complex business logic on records), Reports (scripts that generate reports), Sorting, Printing, Searching, Messages (scripts that display generic dialog boxes to the user), Miscellaneous, and Help (scripts that handle the contextual help system of the program). Such a group of headings works for me; perhaps something else will work for you. But regardless, keep your scripts organized. When there are dozens of scripts, it will make it easier to find the one you want.

If you take another look at the screenshot above, you'll see that some of the scripts are named differently from others. Some of the scripts are indented with spaces and some have part of the name in parentheses and brackets.

In my development standards, scripts that are indented are those that are generally called by other scripts. And, generally, they appear directly below the script that calls them. So you can see from the script list that `Check Layout Records` is called by `Developer Setup`, which also calls `Open Routine`. `Open Routine` in turn calls `Record Logged In User ID` as well as two other scripts.

Parentheses, however, mean that that script takes a parameter, and the words in the parentheses are the names of the parameters. Sometimes, a script can take a parameter but doesn't have to have one. In that case, the parameter name is also placed with brackets to indicate that it is optional. So in the figure, you can see that `Set Window( [PrintFlag] )` is a script that takes one optional parameter, while `Set Next Serial Value( WhichTable, NextSerialValue )` takes two parameters, and both are required.

Finally, every script I write calls a script called `Script Start`. Also, every script I write begins with a comment that documents what the script's purpose is and other useful information. Because I don't want to put these two things into every script manually, I have a `Script Template` script which is never itself called, but is always the start of a new script. When I wish to have a new script, I never create it from scratch. I always create it by duplicating the `Script Template` script and then change the comments to those appropriate for the script.



I think this may have been the most advanced column yet in this series, and I hope that I've done a good job with this tutorial. Take some time to absorb this before we move on next time to more practical applications of scripts in FileMaker. Until then, happy FileMaking!

The Clayton's Web

What's the single most important feature of any book? You'd be right if you answered: the reader, the human being who reads what's on the page.

Book = (paper + ink) + (reader)

OK. That question was easy; now let's try a slightly harder question. What's the single most important feature of a Web site?

Did you get it? The answer is: the visitor.

Web site = (server hardware + software) + Internet + (visitor)

However, let's just break that down a bit more. Here's the reader of the book:

Reader = (eyes + brain)

Here's the Web site visitor:

Visitor = (client hardware + software) + (eyes + brain)
AND/OR
Visitor = (client hardware + software) + (ears + brain)
AND/OR
Visitor = (client hardware + software) + (fingers + brain)
AND/OR
Visitor = (search engine hardware + software)

The visitor to your Web site will in all cases be some computer-powered hardware and software, such as a cellphone, new or old computer, PDA or search engine, and will often, but not always, have a human being behind it.

That human being will interact with the computer hardware and software in various ways, including receiving output by reading things from a screen or paper, listening to a screen

reader, reading Braille, and providing input with any one or more of a pointing device such as a mouse or trackpad, keyboard, or other mechanism.

Once you understand the Visitor, it's easy to notice that some visitors will *see* what's on your Web site, while many will *see and interpret* only the coding behind it. And that's why the coding is so important. Which is why Apple's iWeb is such very disappointing software.

Sound and Fury Signifying Nothing

My experiments with iWeb have produced some quite pretty results. Pages look good, and they are certainly easy to produce. There's nice integration with iPhoto and so on, and the Inspector makes it easy to change fonts and colors and whatnot. Layout's a breeze: just drag and drop. iWeb even uses those nifty little guides that pop objects right into a suitable alignment.

Once you're done, you can press a button or choose a menu command to Publish or Export your files and, heck, they even pass the World Wide Web Consortium's Validator test. The code is valid—it follows the grammar laid down for Web pages.

But what you've created is a pit of seething ugliness, overlaid with the thinnest patina of gorgeous gloss and glitz: proof positive that what's syntactically correct may still be meaningless. It's no more than a Hollywood illusion: it looks like the Starship Enterprise, it sounds like the Starship Enterprise, but it sure isn't capable of lifting off planet Earth, let alone traveling at warp speed.

iWeb Is Not For Web Sites

If all you want to do is create a Web site to show off photos of your dog to your immediate family, and they all have decent computers, decent Internet connections, good eyesight, and are able bodied, then iWeb will do a reasonable job for you.

If you're a professional of some kind then using iWeb to create a Web site is rather like letting a five-year-old nephew design your brochure, letterhead, and Annual Report.

The problem is not only that the code behind the scenes is a mess, but that iWeb provides no way (that I could find) within the interface, to apply some of the most basic rules for making a Web site. And those rules aren't just pedantic puffery; they exist for good, practical reasons.

Headings Allow Skim Reading

You'll see this article is broken up into sections, each with a heading. Headings let the sighted visitor skim through to pick up the gist and zero in on any particularly interesting sections. A blind person using screen reader software can press a key and the software assembles a list of headings so she can skim through and zero in on interesting content. A search engine, such as Google, is like a blind visitor as it doesn't *see* the finished page—it looks at the coding, just like a screen reader does. Google thinks headings are pretty important.

Headings Aren't Big and Bold

Let's just be clear: making text big and bold doesn't make a heading on a Web page—it just makes text big and bold. A sighted visitor sees a short stretch of big, bold text with a certain amount of white space around it and leaps to a conclusion that it's a heading. For software such as a screen reader or a search engine to "see" a heading, the code behind the scenes must specify it's a heading, by using `h` tags, like this:

```
<h2>A fairly important heading</h2>
```

There's no way in iWeb to apply a "heading" style to text with `h` tags—the only option is to make text bigger and bolder, and that isn't a heading; it just looks like one to some visitors.

When Images Go Walkabout

Even more fundamental than headings is alternate text, also known as `alt` text. It's a basic rule of Web coding that when you include an image (and certain other things) you must also put in some text that can replace the image if it goes missing for some reason. There are some comprehensive guidelines about how to choose good alternate text for certain circumstances, but the basic principle is easy: include some text that can replace the picture.

The reason behind this rule is that blind people, search engine software, and those who use equipment that can't or won't display images need some words to tell them what they're missing. Without those words to replace the pictures a Web site can become a nightmare maze of meaningless meanderings. One New Zealand bus company's Web site invites you to click on the link for timetables in your area. If you can't see the pictures they've used for the area names there is no way to know what to click on. In 2000, the Sydney Olympic Committee failed to provide alternate text for the Web site for the games. They were found guilty of discrimination and fined.

Well, iWeb gives us no way at all to add or edit the alternate text for photos we may include.

Getting Your Hands Dirty

The only way you can possibly add correct headings or alternate text is to go in to the coding itself, outside of iWeb, and using a text editor. If you know how to do that, then you wouldn't want to wade through the tangled mess iWeb creates. And you'd have to do it over every time you changed something on the page.

iWeb For Fun, But Not For Profit

iWeb's a fun product. It's easy to drag and drop text and photos into places on the page. If it had an export-to-PDF function it would probably be pretty good for creating printed materials.

If you have a clearly-defined audience and know their capabilities, if you don't care about search engines, if you don't need to consider those on slow connections or who might have any kind of special needs, if you're creating a small family-type Web site, then iWeb may be fun to play with.

If you are in any way publishing to a wider, largely unknown audience, or have any kind of even slightly professional texture to your site, then iWeb is most definitely not for you. The software doesn't just create extremely poor coding, but it *prevents* you from implementing some of the most fundamental principles of an acceptable Web site.

A Disappointment

It's not like the notions of headings or alternate text are in some way *new*—they've been there in the rules of HTML since the beginning. As a Web designer and Mac user I feel so disappointed in Apple. This software is gorgeous, and shallow. It's likely to set back what we're trying to achieve with the modern Web: access for all to fun, information, people, and community.

The Clayton's Web

In New Zealand in the 1980s there was a lot of concern about drinking and driving. A company produced a non-alcoholic beverage called Clayton's that resembled Scotch. There was a big advertising campaign that backfired: "Clayton's—the drink you have when you're not having a drink." Clayton's came to be a catchphrase for something that wasn't what it purported to be, a sham, a fake.

The real Web is a realm of freely available information: the coding behind the scenes caters to all equipment, all visitors. The Clayton's Web looks good, maybe even looks fantastic, but in fact it's a sham, a disappointment.

iWeb: it's the Web software you don't use when you're making the Web.

Related Articles

- Web Accessibility, Part 1, ATPM 10.01, January 2004.
- Putting Curb Cuts and Wider Doors on the Internet: Toward Web Site Accessibility, ATPM 8.11, November 2002.

How To

by Sylvester Roque, sroque@atpm.com

Maybe You Ought To Be Using Automator

> "What Automator does is it gives you all the power and ability of AppleScript, and the other scripting languages built into the operating system, but in a drag and drop format, where you're not actually writing anything you're just checking boxes, radio buttons, pop up menus, that kind of thing to determine what you want to do."
>
> —Sal Saghoian, Automator product manager, Apple Computer

I came across that quote, which appeared in a May 2006 MacBreak podcast, while researching this article. Mere days before that, I had decided for the second time to try to learn a little about Automator as part of my resolve to do things on the Mac this year that I have never done before.

I had programmed a smidgen in Applesoft BASIC years ago. It didn't go badly, but it did convince me that I didn't like coding. I tried AppleScript but didn't have many applications that supported it at the time and found it a little confusing. Needless to say, that didn't last long either. My last programming effort actually resulted in a semi-working series of macros in Microsoft Excel.

With the advent of Tiger came Automator, a drag-and-drop programming language. I tried it a few months ago for all of about five minutes, but maybe it's time I lived up to my resolution and learned something new. If you think this is complicated or that you can't create something useful, let's dive right in and see what happens. There are a few things you need to know before we begin.

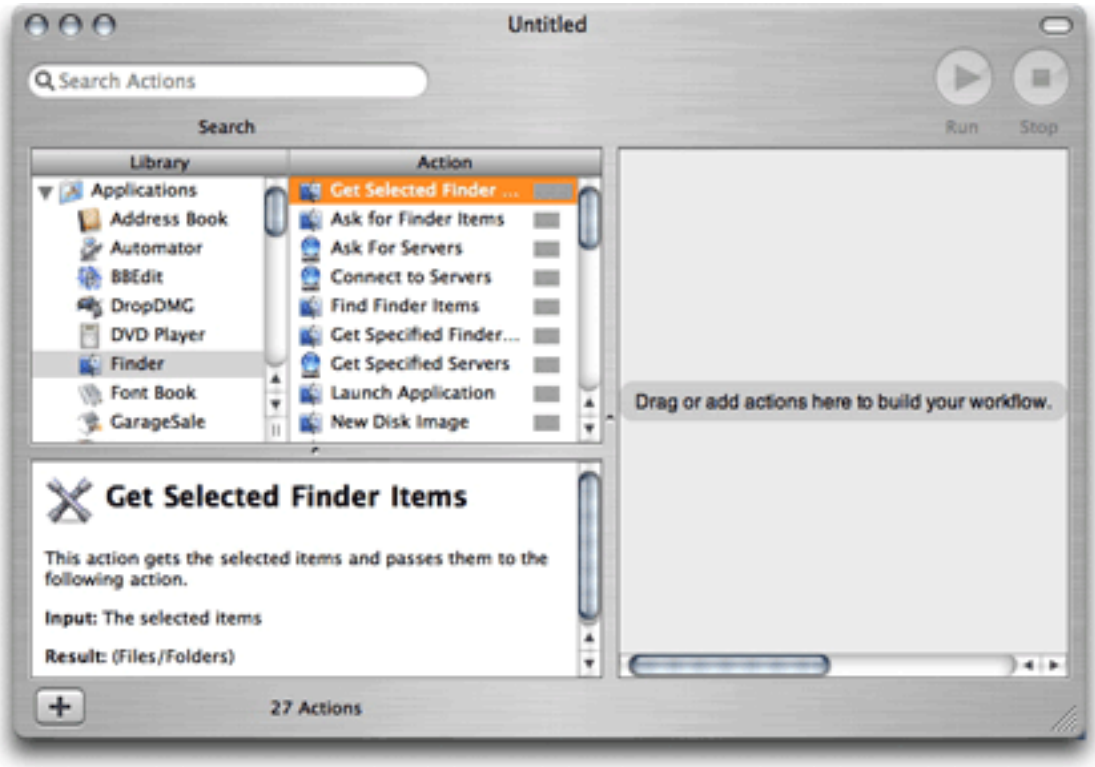
Before We Begin

A growing number of applications are "Automator aware"—they contain pre-defined steps or "actions" that Automator knows how to execute. If you have ever looked at AppleScript dictionaries, this is a similar concept. Actions are the individual steps that must be executed for an application to finish a task.

In the real world, if you put the right steps (actions) in the right sequence, you complete a task. In Automator, the sequence of steps needed to complete a task is called a "workflow." In other languages or programs, these workflows might be referred to as "macros" or "scripts." Of course, workflows can be saved, or they wouldn't be very useful to you. There

are several options for saving your workflows, but we'll get to that in a minute. For now, let's get to work by taking a look around Automator.

Open your Applications folder and look for Automator. You'll recognize it by the icon of a robot carrying what appears to be a bazooka. It seems that many have started to call our little robotic friend "Otto." And no, that is not a bazooka in his hand. It is, in fact, a pipe—representing the power of the Unix pipeline. Those of you with a better understanding of Unix than I have will understand what that means. In the MacBreak podcast I mentioned earlier, Leo Laporte suggested that Automator workflows might be a good way to give users a GUI for languages such as Perl that would normally require setting several parameters.



Once launched, the basic Automator window is fairly easy to navigate. In addition to the typical close, collapse, and expand controls, the upper left portion of the window contains a search box. Enter a keyword here and Automator will search the actions installed on your system looking for those that match what you want to do. Search results are ranked based upon their relevance to what you seem to be doing at that moment in the workflow. Entering the same keyword at different times in the workflow might result in a slightly different ranking of results.

The upper right portion of the window contains Run and Stop buttons. Workflows are usually run without having to open Automator, but we'll get to that in a moment.

The remainder of the window is divided into three panes with the leftmost pane subdivided into two vertical sections. At the far left is a list of all "Automator aware" applications currently installed on your hard drive—including the Finder and some basic system-level tasks. The very bottom of this pane contains a folder of example workflows and a folder of workflows that you have created.

The second vertical pane lists Automator actions. Selecting a specific program icon in the left pane causes the actions listed here to change, reflecting the actions available for the program you selected. Try it and see what happens. Single-click the Finder icon in the left pane. Now you see a list of actions that the Finder understands. When you search for a function, your search results appear here as well.

While clicking on some of the Finder actions, you may have noticed that the text in the smaller pane at the bottom of the window changes. Click an action and its description appears here. Although these descriptions are written with the non-programmer in mind, some are easier to understand than others.

The dominant pane in the Automator window is to the far right of the screen. Workflows are built by dragging actions from the center pane into this area. Generally actions are dragged here in the order that they are needed in a workflow. If you skip a step, don't fret. Drag the missing step where it's needed and drop it in place.

Now that we've looked at the basics, let's build some potentially useful workflows. After the basic workflows, we'll tweak them a little to cover a wider range of tasks. We're going to back up some music and set up a work environment that opens after each login.

Our First Project

When was the last time you backed up that gargantuan music folder? Let's take care of that right now. We'll use the first workflow I ever created to back up a specific folder. I built this to back up my music collection, but it can back up any folder you choose. One word of caution: until you are sure it's working properly, either have a backup handy or use sacrificial data. Before turning it loose on my music collection I used a test folder with a few small files.

Before we drag the first action into place, let's think about what happens when we manually copy a file or folder using the Finder. You must locate the source file and click to select it, drag the file to its destination, and drop it in place. If there's already a file present with that name, you must tell the Finder whether or not to replace the existing files. We're going to use these steps to guide our search for the right Automator actions. With that information in hand, as we complete each step see if you can guess the next step.

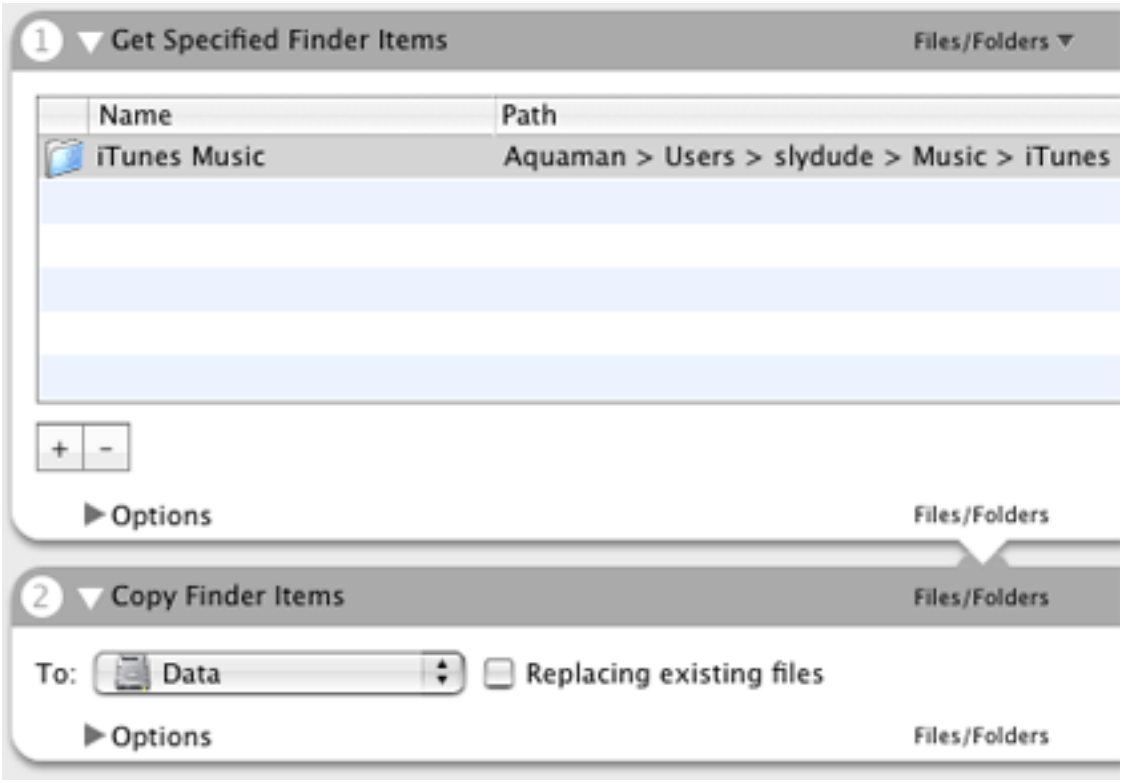
Since we will be using the Finder to copy the files, click "Finder" in the left hand pane. We need the name and location of the file or folder to copy, so look in the middle pane for an action called "Get Specified Finder Items." This is how we will tell the Finder what to copy. Drag that action into the rightmost pane and drop it in place. Double-clicking the action will also move it into place.

If this action appeared in the rightmost window but didn't expand to show a way for selecting files and folders, click the white arrow at the top of the action. Automator uses that white arrow to indicate that there are parameters available for the user to adjust.

The expanded view of this action contains a plus and minus sign. Click here, and the standard file selection sheet appears, allowing you to chose files or folders to copy. They will appear in a list inside the action. If you accidentally add something you don't want to copy, select that file in the list and click the minus sign to remove it.

Before we leave this action, click the triangle labeled "Options." Now a choice appears which says "Show Action When Run." If you are going to copy the same files all the time leave, this option unchecked. I did that with my music workflow so it copies the same folder each time and runs unattended. If you want to select different files each time the workflow is run, check this option. The file selection sheet will now appear each time the action is run.

Now that we have selected the target files, what's the next step? That's right: tell the Finder to copy the selected files. This action has a pop-up menu to choose where the new files should be placed as well as a checkbox that allows us to tell it whether to replace existing files. The action has additional parameters accessed by clicking the Options triangle, but for now we we'll leave those as they are.



Run the workflow once to make sure it works as expected. If the files have been copied correctly, save your workflow. Choose Save As from the File menu, name the file, and save

it as an application. Right now this workflow runs when I double-click it. After we finish the other workflow, I'll show you how to use an iCal alarm to trigger this little application to run at a specified time.

This workflow is an efficient means of copying folders to a new location. There are times, though, when you may want to copy only files that have been modified. To do that we would have to teach Automator a new trick, so we'll save that for later.

Getting Right Down to Work

When I log into my computer, the first two tasks I perform are checking e-mail and visiting several Mac-related Web sites. Currently, I launch Mail and Safari and click bookmarks for the sites I want to visit. I could have Mail and Safari automatically launch by adding them as login items for my account, but that doesn't get me to the sites I want to visit without some intervention on my part. I would still have to type in the names of sites I want to visit, open a new window, type in a new URL, and repeat the process until all the target sites are loaded. A boring and repetitive task—sounds perfect for "Otto."

I want the Safari-related actions to execute first and Mail-related actions to occur last. This puts the mail window on top and I can read incoming mail before I start Web browsing. Let's think a bit about the necessary steps before we begin. That should make finding the right actions much easier. As far as I can tell the steps have to occur in this order:

1. First, I Launch Safari.
2. Safari needs the names of the sites I want to visit. More accurately, Safari needs the URLs for specific sites or pages.
3. Safari opens with my home page in a new window. I need to open a new window and add the URL for one of the sites I want to visit and have Safari display that page. I also have to repeat the process for each site I want to visit.
4. When that's finished, I launch Mail. It's already configured to check my accounts upon launch. I also don't need to enter my password. That information is stored as part of my account information.

This time, the first thing I need to do is launch Safari. Normally I would go to the Finder actions and search for one that could launch an application. That would probably work, but I found out that in this case I don't need to launch Safari before passing it the URLs for sites I want to visit. In a moment, I think you will understand why a specific launch command is redundant. Since you did so well with the first workflow, I'm going to go a little faster with this one until I get to any difficult parts.

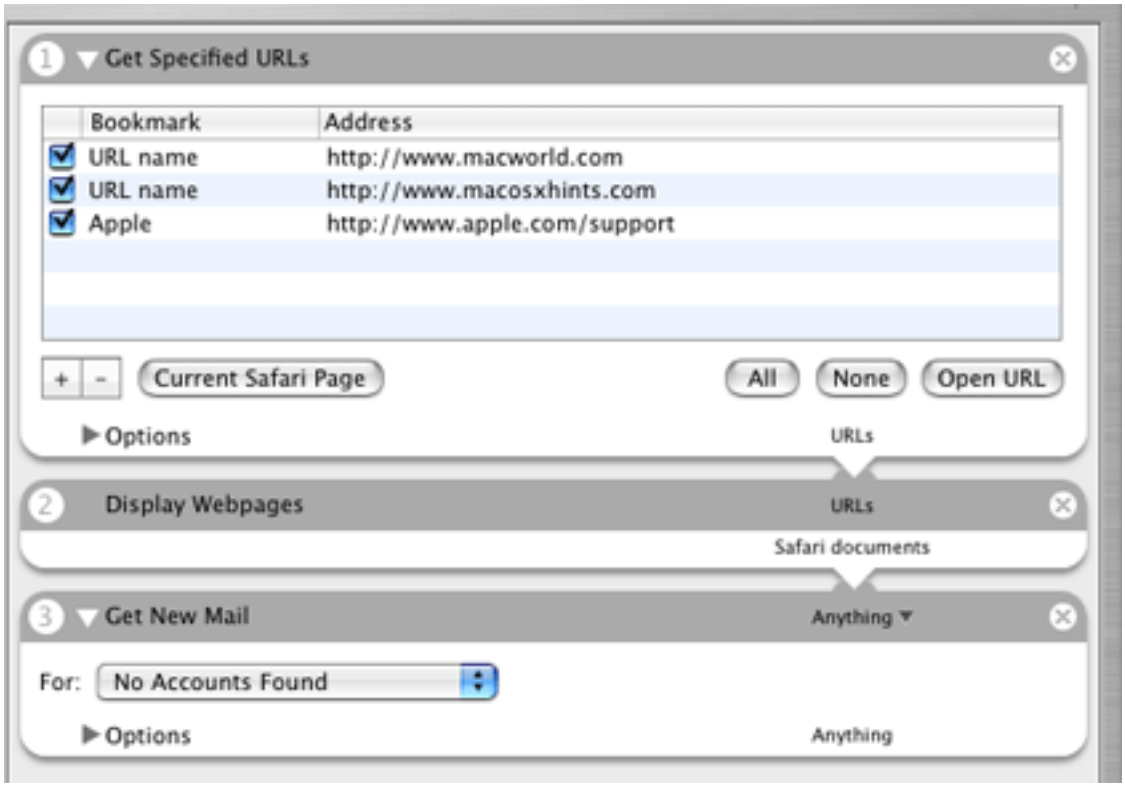
First Otto needs the URLs for my favorite pages. Since it's Safari that needs the specific URLs, I'm going to look there for the first action that I need. A quick glance and I see "Get Specific URLs." Dragging it in place I notice that the box allows me to enter multiple

URLs. Make sure you enter the URLs accurately or the pages won't load. With that done, it's time for the next step.

Once you have the URLs in place, you need Safari to show them to you. I don't see a Safari action that says show Web pages, but looking at the list I bet you can guess what I need to do. That's right, I need the "Display Web pages" action. This action results in Safari documents. Safari and Automator are smart enough to know that Safari must be launched before it can create the documents. With that action in place, I don't need a specific launch command. That's taken care of for me.

Now that we are done with Safari, our next task is to get any new mail. We are looking for an action that will retrieve (get) new mail. Lucky for us, Otto has that action in the Mail section. Find that action and drag it into place as the next step. Don't forget to set the pop-up menu that tells Mail which accounts to check for new mail. The options for this action allow checking all accounts or checking a single account. Set the options to meet your needs and save the workflow as an application. I set the application as a login item in System Preferences. I'll describe how to do that in a moment. On average Automator, does these tasks eight seconds faster than I can every time I log in.

You've kept up with me thus far, so here's a pop quiz. The "Get New Mail" action will either check one e-mail account or check all accounts. How do you check multiple accounts without checking all of them? No, I don't want to run the entire workflow several times. You guessed it. Add multiple instances of the "Get New Mail" action with the pop-up menu in each one set to check a different account. Mail will only launch once, but each account you specify will be checked.



There are two things to keep in mind when you run this workflow. First, there is no error checking. If you don't have an active Internet connection or there are other problems, Safari and Mail generate the same error behavior that they normally would.

Second, this workflow forces Safari to open pages in new windows even if you have enabled tabbed browsing. There is an action to open Web pages in tabs. Simply install this action and use it in place of the Display Web pages action. Kudos to fellow ATPM editor Eric Blair for a job well done.

There is an alternative action to open Web sites in tabs. It does require you to build a folder with Internet Connection Files for sites that you want to visit. Once you create the folder and set up the workflow, don't move or delete that folder. If you do, the workflow will stop working until you either create a new folder or tell the workflow where to find the folder.

Making the Workflows More Useful

What could we do to make these workflows more useful? How about making them run when we want them to or with little or no user intervention. I'll briefly share a few ideas?

Let's look at the workflow we used to copy files. Save copies of this workflow as applications with descriptive names to help you remember what's being copied. Don't forget to change which files or folders are being copied. Run these workflows to copy different sets of targeted items with no more user intervention than a double-click. On one copy, go to the "Get

Specified Finder Items" action and click the Options triangle. Click the "Show Action When Run" checkbox. This box will now appear every time the workflow is run, allowing you to choose different sets of files each time the workflow runs.

If you have a large amount of data to copy, set the workflow to execute at a more convenient time, like late at night. Let Automator and iCal work together. First you need to open your copy files workflow and modify the "Get Specified Finder Items" action so that it targets files that you want to copy. If necessary, deselect "Show Action When Run." From Automator's File menu choose, Save As Plug-in. Enter a descriptive name and save it as an iCal plug-in. Launch iCal and set an alarm for the time you want your workflow to execute. When you set the type of alarm, choose "Run Script." Just beneath that is an option that will now say Script and None. Click where it currently says "None" and choose "Other" You will be given an opportunity to locate the script you just saved as a plug-in.

Saving a workflow as a plug-in allows it to be used as part of a specific program. Saving the previous workflow as an iCal plug-in makes it available to, and behave like, part of iCal. You can save plug-ins as part of other programs as well.

Initially, I wasn't too impressed with the ability to save workflows as plug-ins, but I have changed my mind. When fellow editor Eric Blair saw a draft of this article, he pointed out that saving workflows as Finder plug-ins allows users to create their own contextual menu items. Imagine being able to right click files and upload them to a specific server, or create disk images with specific settings by working in conjunction with DropDMG (a utility from ATPM's publisher).

Sometimes you might want to compare two folders and synchronize them. This could be very helpful when copying large amounts of data. In looking at the actions that shipped with Automator, I didn't find anything that would make this possible. Fortunately there is an action available called compare folders that does just that. Once you download and unzip the file, choose Import Actions from Automator's File Menu and locate the unzipped action. From now on, you can use this action in your workflows.

Consider saving your workflows as either applications or plug-ins. Workflows in either of these formats can run without needing to have the Automator application open.

Just like teaching an old dog new tricks, you can teach Automator new actions. The actions then become available in any of your future workflows. If you are feeling adventurous you can write your own actions in AppleScript or any other scripting language that the operating system understands. For mere mortals like the rest of us, there's always the Automator actions at AutomatorWorld, or this download page.

I have only scratched the surface of what Automator can do. Other users have written workflows for a wide variety of repetitive tasks such as grabbing all the pictures on a Web page. My next project is a workflow that will copy files from a work-in-progress folder to a memory stick drive. I'd like to have the files get copied while I am asleep, log out of my account, and put the computer to sleep. I'll keep you posted.

Desktop Pictures

Alaska

This Month's Desktop Pictures

This month's photos of Alaska are courtesy of John Lowrey at Northern Softworks, a Mac shareware and freeware developer since 1998.

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

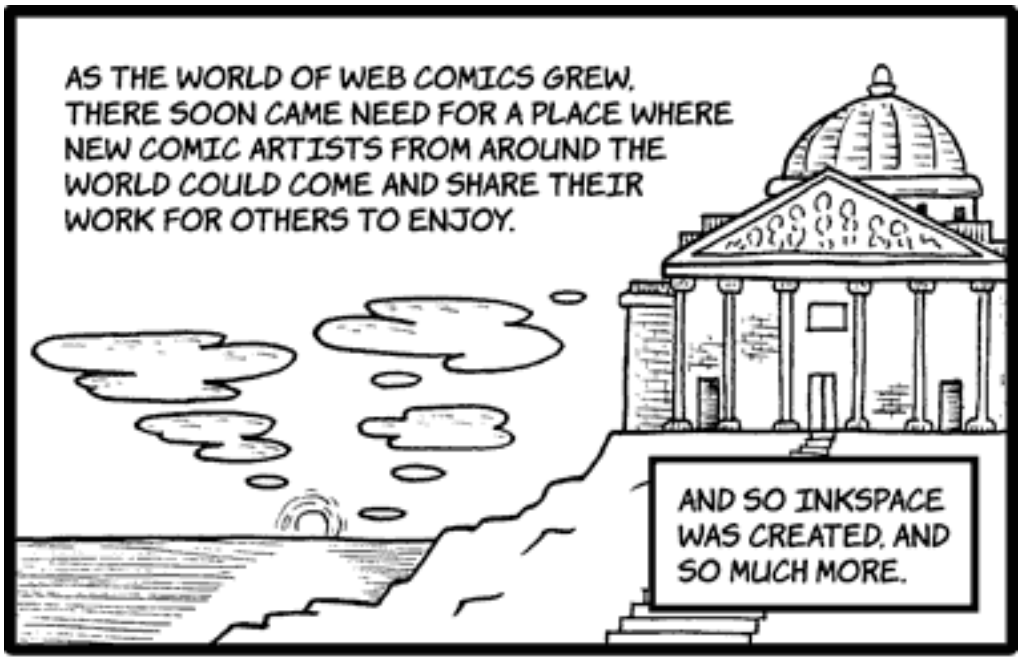
the ATPM pictures in your Pictures folder, you're all set. Otherwise, click Configure to tell the screen saver which pictures to use.

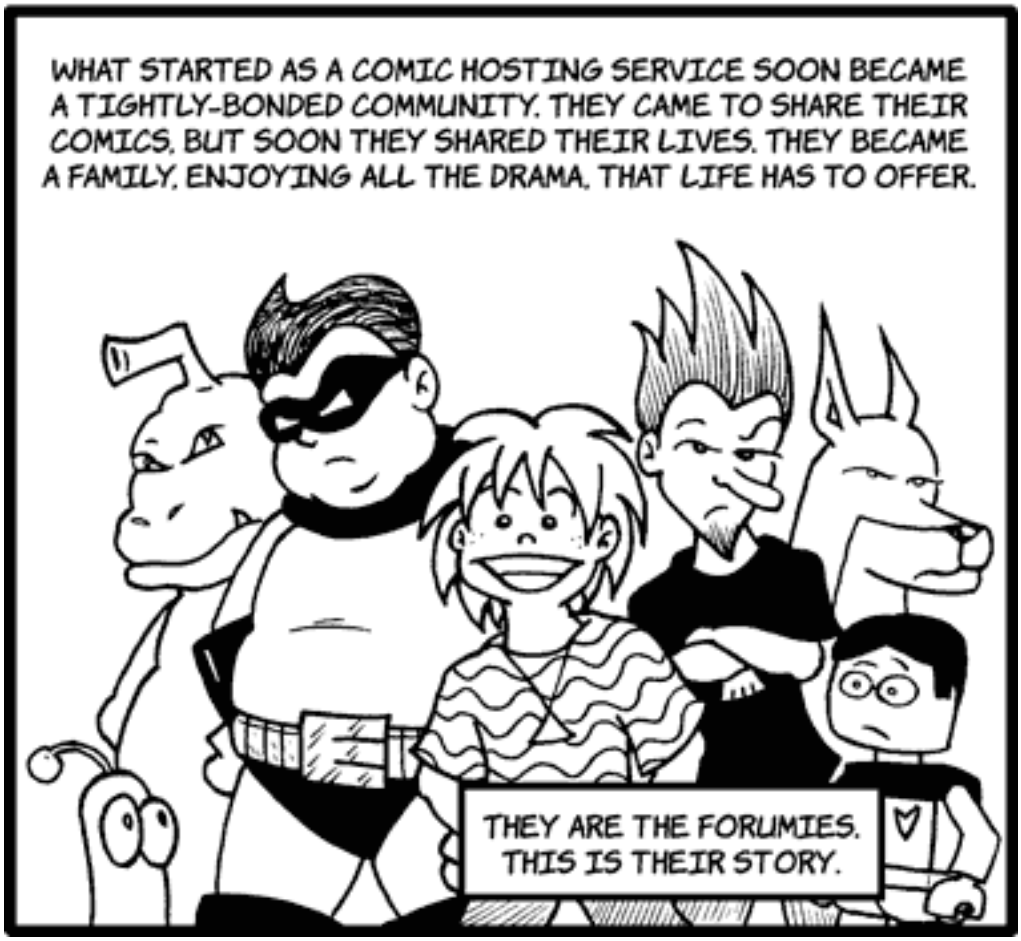
Mac OS X 10.0.x

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.

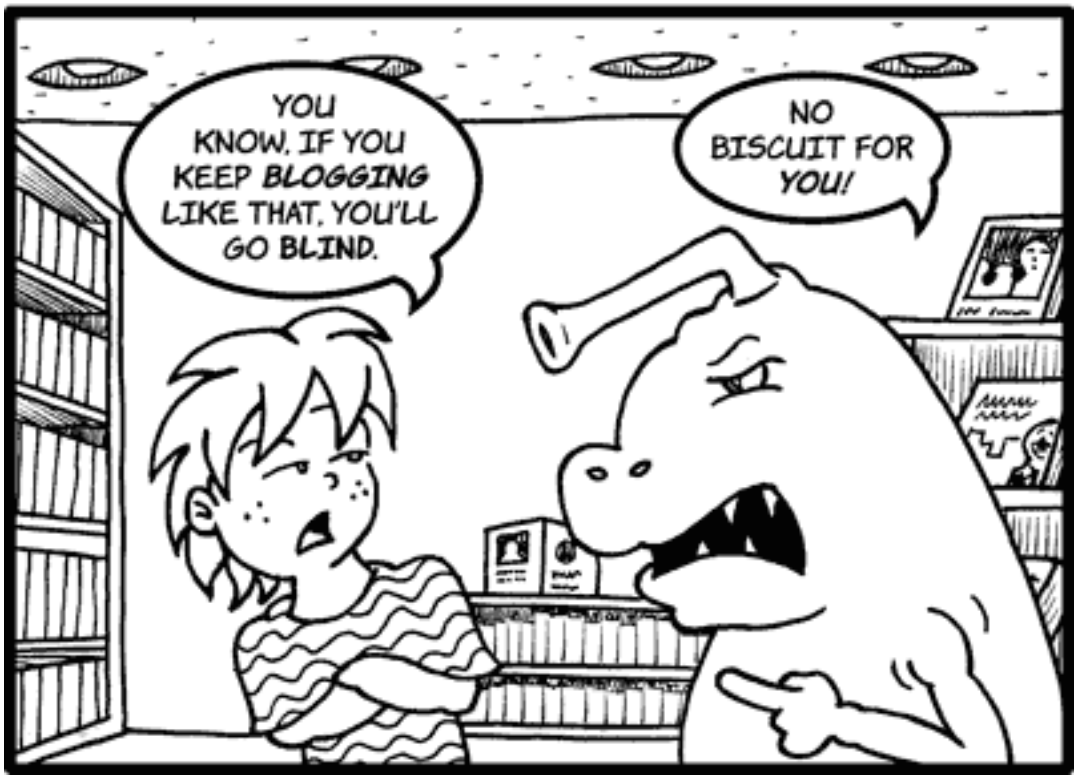
Cortland

by Matt Johnson, mjohnson@atpm.com


AS THE WORLD OF WEB COMICS GREW, THERE SOON CAME NEED FOR A PLACE WHERE NEW COMIC ARTISTS FROM AROUND THE WORLD COULD COME AND SHARE THEIR WORK FOR OTHERS TO ENJOY.
AND SO INKSPACE WAS CREATED, AND SO MUCH MORE.

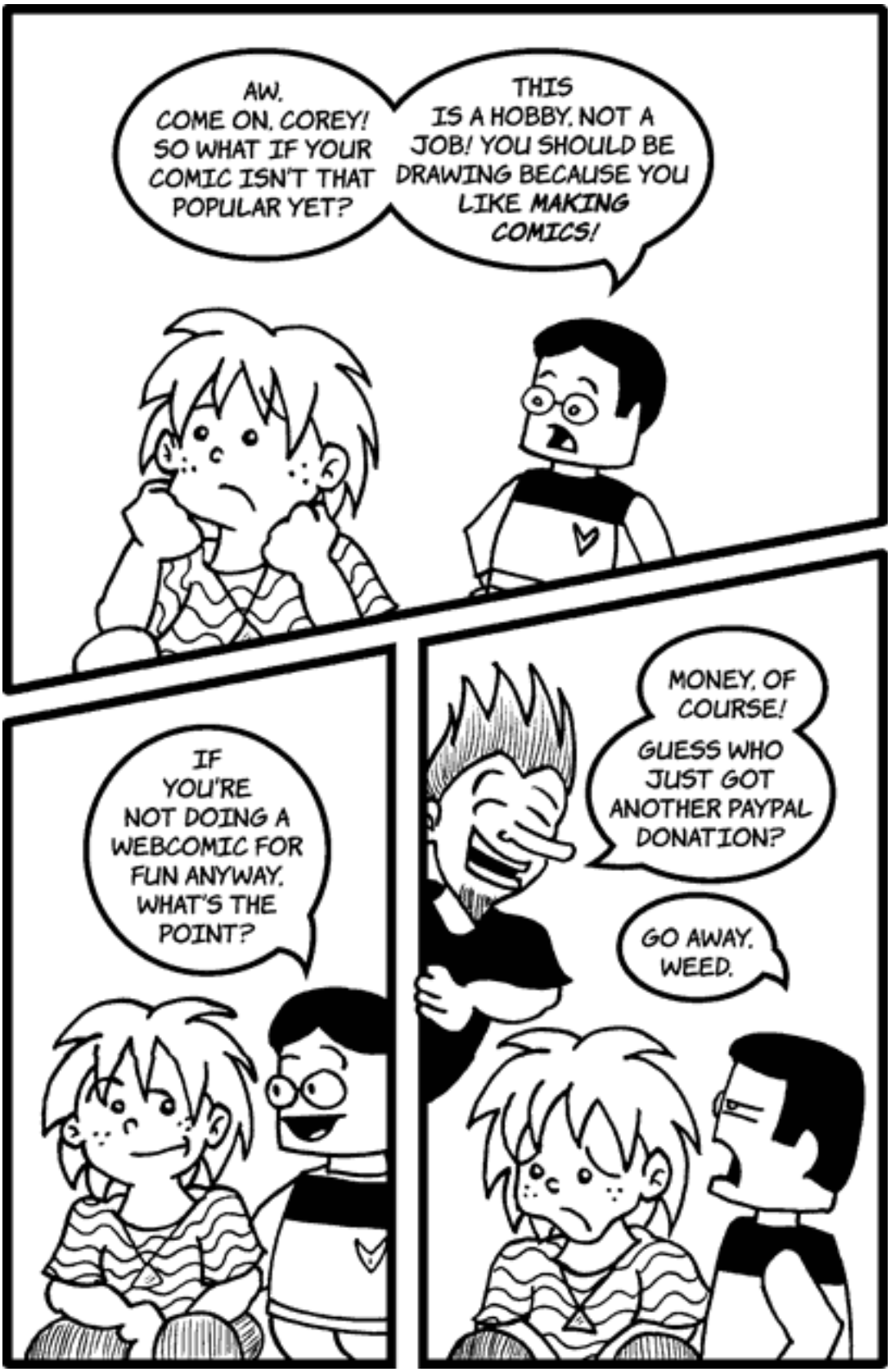

WHAT STARTED AS A COMIC HOSTING SERVICE SOON BECAME A TIGHTLY-BONDED COMMUNITY. THEY CAME TO SHARE THEIR COMICS, BUT SOON THEY SHARED THEIR LIVES. THEY BECAME A FAMILY, ENJOYING ALL THE DRAMA, THAT LIFE HAS TO OFFER.
THEY ARE THE FORUMIES. THIS IS THEIR STORY.

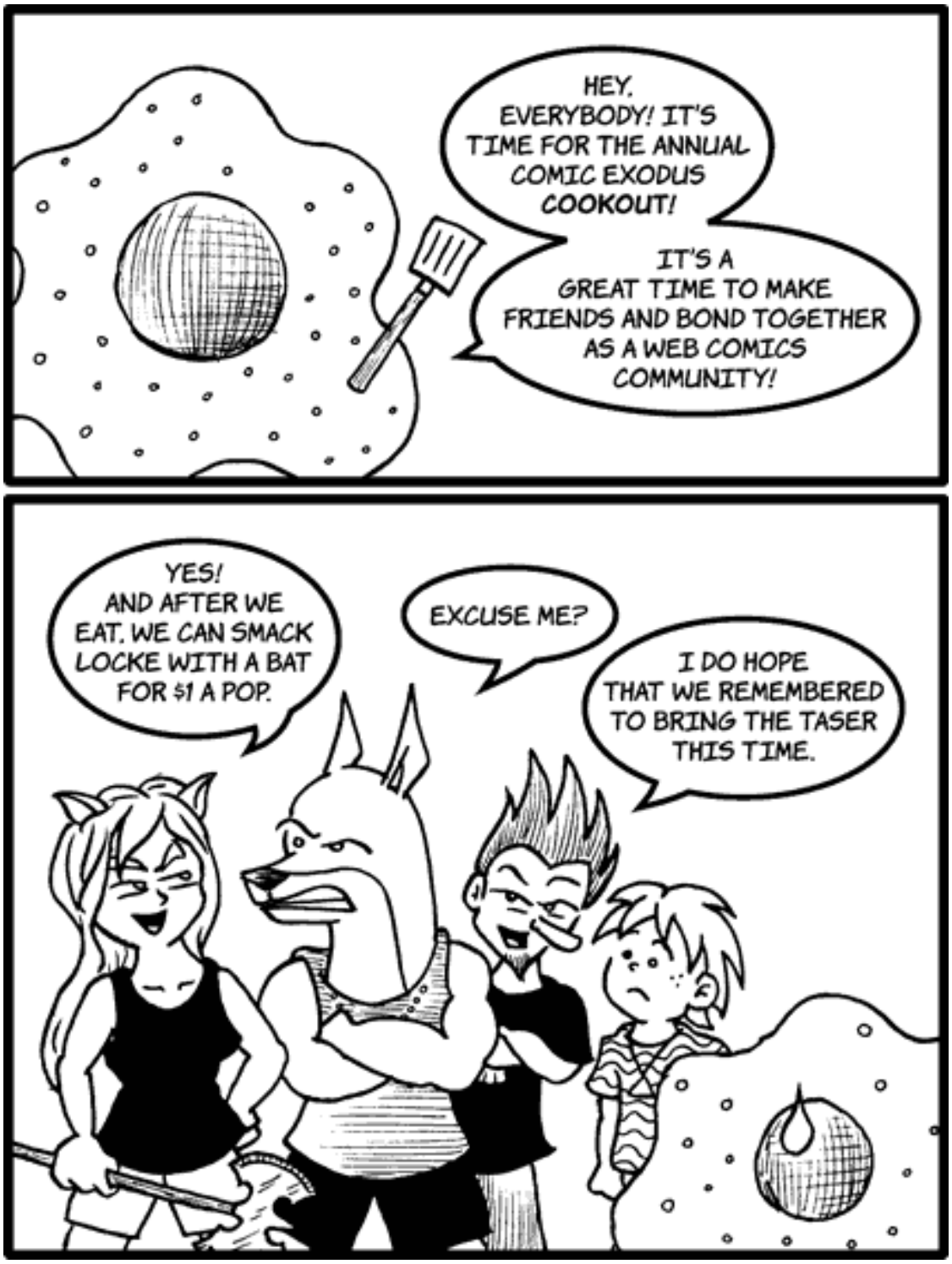

HI, EVERYBODY! I'M THE WEBSNORK! I'M A WRITER, AND I'M TOTALLY HOT SHIT THESE DAYS!
DON'T BELIEVE ME? CHECK IT OUT: I WROTE THE FORWARD TO CAPTAIN EGO, VOLUME 3, AND THIS BOOK IS ACTUALLY AVAILABLE IN SELECT BARNES AND NOBLE BOOKSTORES!
...AND IF THAT DOESN'T MAKE ME A LEGIT WRITER-TYPE PERSON, NOTHING DOES!
CaptainEgo VOLUME 3


YOU KNOW, IF YOU KEEP BLOGGING LIKE THAT, YOU'LL GO BLIND.
NO BISCUIT FOR YOU!

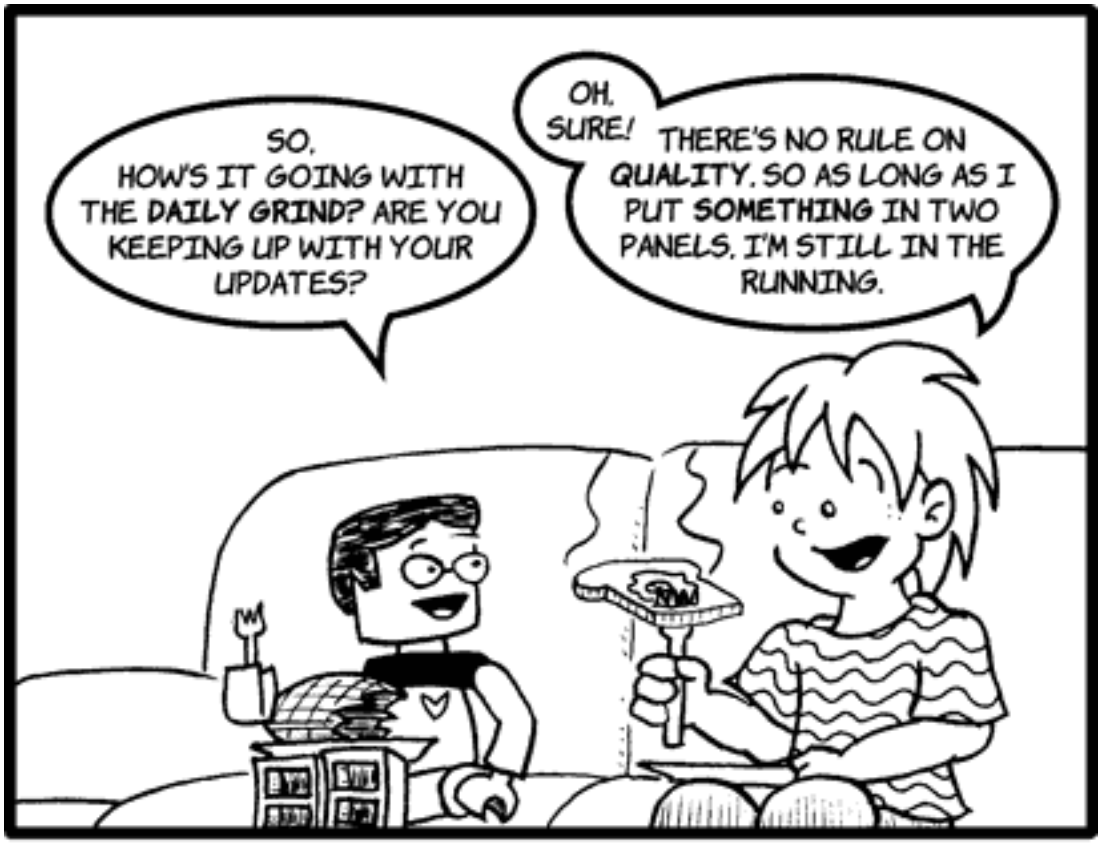

SO HOW'S THE COMIC COMING, COREY?
MAN, IT'S REALLY TAKING OFF! MY READERSHIP'S DOUBLED IN THE LAST MONTH!
OH, REALLY?
YEAH! I USED TO HAVE ONE UNIQUE VISITOR. NOW I'VE GOT TWO!


HEY, COREY. WHY DON'T YOU SEE HOW YOU RANK AGAINST ALL THE OTHER COMICS HERE ON THE THE GUIDE?
HEY, YEAH!
NO, COREY! DON'T DO IT!
AUGH!
THAT WAS MEAN, WEED.
I'M LESS POPULAR THAN A STATIC IMAGE OF MEGA MAN'S HEAD.


AW. COME ON, COREY! SO WHAT IF YOUR COMIC ISN'T THAT POPULAR YET?
THIS IS A HOBBY. NOT A JOB! YOU SHOULD BE DRAWING BECAUSE YOU LIKE MAKING COMICS!
IF YOU'RE NOT DOING A WEBCOMIC FOR FUN ANYWAY. WHAT'S THE POINT?
MONEY. OF COURSE! GUESS WHO JUST GOT ANOTHER PAYPAL DONATION?
GO AWAY. WEED.


HEY, EVERYBODY! IT'S TIME FOR THE ANNUAL COMIC EXODUS COOKOUT!
IT'S A GREAT TIME TO MAKE FRIENDS AND BOND TOGETHER AS A WEB COMICS COMMUNITY!
YES! AND AFTER WE EAT, WE CAN SMACK LOCKE WITH A BAT FOR $1 A POP.
EXCUSE ME?
I DO HOPE THAT WE REMEMBERED TO BRING THE TASER THIS TIME.


MAN, THIS IS THE WEIRDEST ICED TEA YOU HAVE HERE!
HOW SO?
IT'S LIKE REGULAR TEA, BUT IT HAS ICE IN IT!
NO HIGH FRUCTOSE CORN SYRUP AT ALL, THEN?
YEAH, WEIRD!


SO, HOW'S IT GOING WITH THE DAILY GRIND? ARE YOU KEEPING UP WITH YOUR UPDATES?
OH, SURE! THERE'S NO RULE ON QUALITY. SO AS LONG AS I PUT SOMETHING IN TWO PANELS, I'M STILL IN THE RUNNING.


GOKU AND I WILL DEFEAT YOU!!


NO I AM DARTH VADER

Software Review

by Charles Ross, cross@atpm.com

DiscBlaze 6.1.6

Developer: RadicalBreeze Software
Price: $20 (normally $30)
Requirements: Mac OS X 10.4, a Mac OS X–compatible CD or DVD burner. Universal.
Trial: Fully-featured (5 burns)

I tend to burn many DVDs, usually a few every week. My need for data storage data has constantly outstripped existing hard drive technology's capacity to store it all. But for the most part, these are just archive volumes, not anything I send to clients or anyone else, so my needs are pretty simple.

My routine goes something like this: I use an AppleScript I've written to create a new 4.7 GB sparse disk image, copy stuff to it until it's full, then burn the image using Disk Utility. Label it with a Sharpie and file it away.

Given the volume of disks I burn, I decided to check out DiscBlaze, an application dedicated to burning CDs and DVDs while providing more advanced options than are available with Disk Utility.

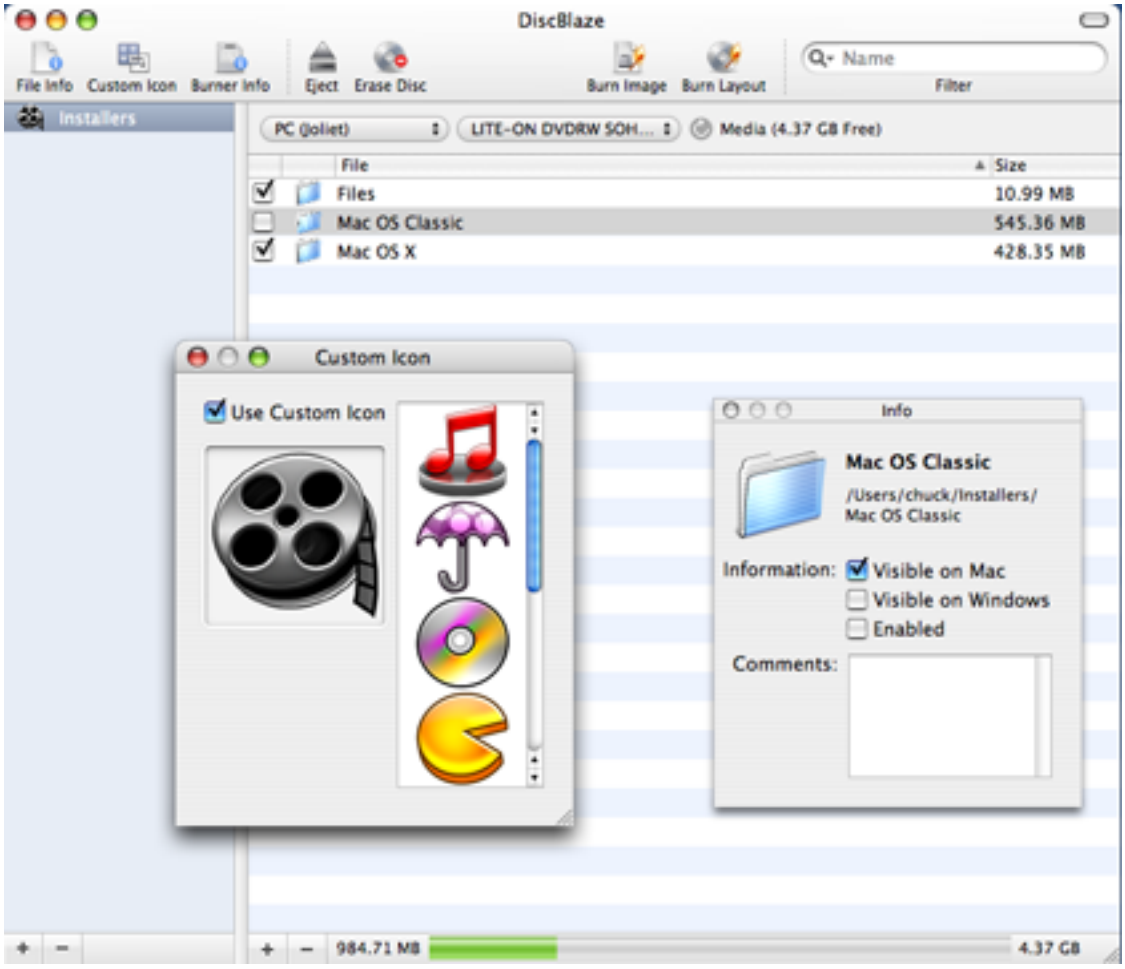
While DiscBlaze seems to be a well thought out and well written piece of software, I'm afraid it didn't really do anything for me. After using it for a while, given the simplicity of my needs, I find it actually more difficult and time consuming than my AppleScript/Disk Utility solution.

For instance, I find using disk images convenient, as the amount of space left is easily determined and files can be copied from the Finder. I don't need multiple formats as only Macs will be reading these disks in the future, so the native HFS+ format of Disk Utility works fine for me. I'm not constantly burning disks, and don't mind if the burn in the background takes a few minutes longer than it would with DiscBlaze, as I'm already doing other things on my computer during the burn.

Since disk images appear automatically in Disk Utility and I have a shortcut that launches the program, burning a disk from an image takes one keystroke and two clicks, and I could probably bring that down to a single keystroke if I took the time to write an AppleScript for the burning portion of the process. I also don't burn Audio CDs often, so iTunes is sufficient for me in that case, and I never burn MP3 CDs, since my iPod is completely sufficient for taking my music with me.

So, I can confidently say that I wouldn't purchase the software at this time. Honestly, the features it offers would seem more useful in the days before software distribution via the Internet. By that, I mean that if I were a software developer who shipped CDs to customers, I could see the utility of DiscBlaze for burning a single cross-platform image many times with a custom icon, but that's not applicable to me. I get software to my customers and clients via e-mail, FTP accounts, and my Web site, without the shipping of physical atoms.

Having said all of that, DiscBlaze does offer an impressive feature list for multiple disk burning options. Disks can be created that will work on just Macs or on Macs and PCs, and the options for burning music CDs seem more comprehensive than those available with iTunes. Some nice custom icons have been provided that can be applied to a disk. A nice bar is shown that gives feedback as to how full the disk is as you add files. When burning to a cross-platform format, you can easily specify which files appear on which platform.

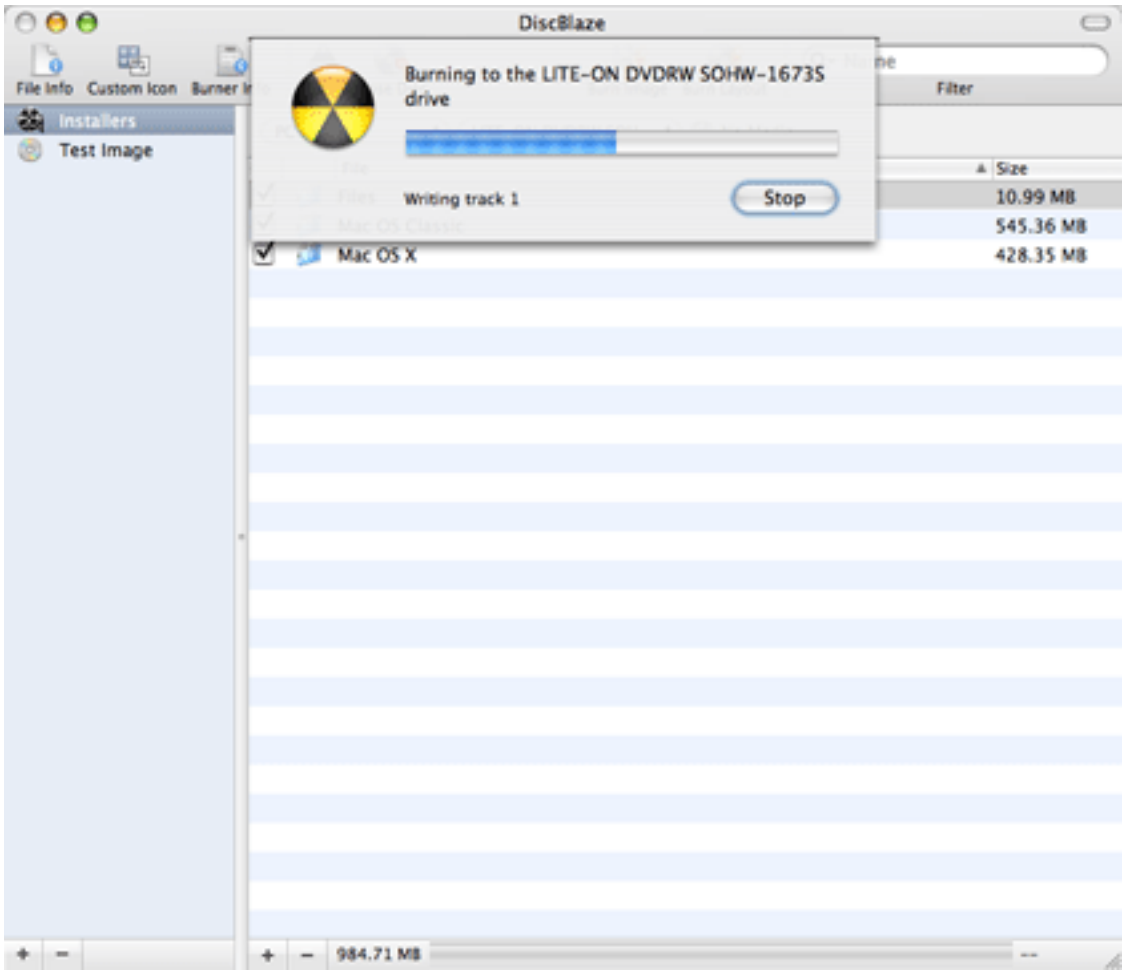


The interface of the software is clean and easy to navigate. Although I've not used the software prior to the current version, this version was recently rewritten using Cocoa, which makes the program feel like a well-behaving Mac OS program (although some additional contextual menus would be nice).

Regarding the speed, I did some tests using an HFS+ sparse disk image with 2.38 GB of data on it. Burning and verifying the disk with DiscBlaze took 10 minutes and 8 seconds on my 1.83 GHz MacBook Pro with an Iomega external DVD burner. Using Disk Utility, the same burn took 19 minutes and 33 seconds. This isn't a perfect test, as I was doing other things during both of the burns, and what I was doing wasn't the same each time, so that may have affected the burn times. Still, that's quite a time difference regardless, which may be important to you if you're burning dozens of disks a day. One nice difference between the two was that DiscBlaze kept the disk image mounted during the burn, whereas Disk Utility unmounts the image during the burn and remounts it afterwards. Of course, this might be problematic if you attempt to write to the image while it's mounted and burning.

I had only one problem with the software performing as expected: I have two DVD burners, and for some reason could not get the software to burn to the MacBook Pro's internal drive while the external one was connected. I could set the internal drive as the target, but when I opted to burn an image, it always showed only the external drive as the destination. Disconnecting the external drive solved the problem.

But that also brings up another possible negative point: concurrent burns. Even assuming that the bug preventing me from targeting the internal drive is solved, the DiscBlaze interface is unavailable while a disk is burning. This means that if I'm burning a disk to one drive, I'm unable to use the program to burn another disk to the other. I have done this on occasion with Disk Utility, which displays burn progress in a separate window rather than a sheet, similar to Finder copy progress windows. It's not a major issue, but it's one area where DiscBlaze is actually less useful than Disk Utility as far as burning is concerned. Of course, this is completely offset by the difference in burn times with DiscBlaze.



If, unlike me, the features offered by DiscBlaze (speed, multiple formats, customization) are useful to you, the software works well at what it does, and it is mature and stable and has a clean and useful interface. Also, at this time, although the normal price for DiscBlaze is $30, it's currently being offered for $20, so now would seem to be the time to purchase it if you find it useful.

Software Review

by Charles Ross, cross@atpm.com

Dobry Backuper 1.5

Developer: dobrySoft
Price: $30
Requirements: Mac OS X 10.3, hard drive space for CD or DVD images when backing up to those media. Universal.
Trial: Fully-featured (30 days)

There's a proverb in IT circles that the world is divided into two types of people: those who have lost data and those who will. (Another one I always enjoyed when my job was to troubleshoot computer systems and networks: if it didn't happen twice, it didn't happen. And, on a more humorous note—the world is divided into 10 types of people: those who understand binary and those who don't.) Everybody at one time or another will find that an important file or, even worse, a critical hard drive, has become corrupted or lost. The reason may be that you emptied the trash when you shouldn't have, or that the drive had a mechanical failure (those darn atoms!). Regardless, if you haven't already lost some data that you wanted to keep, you will.

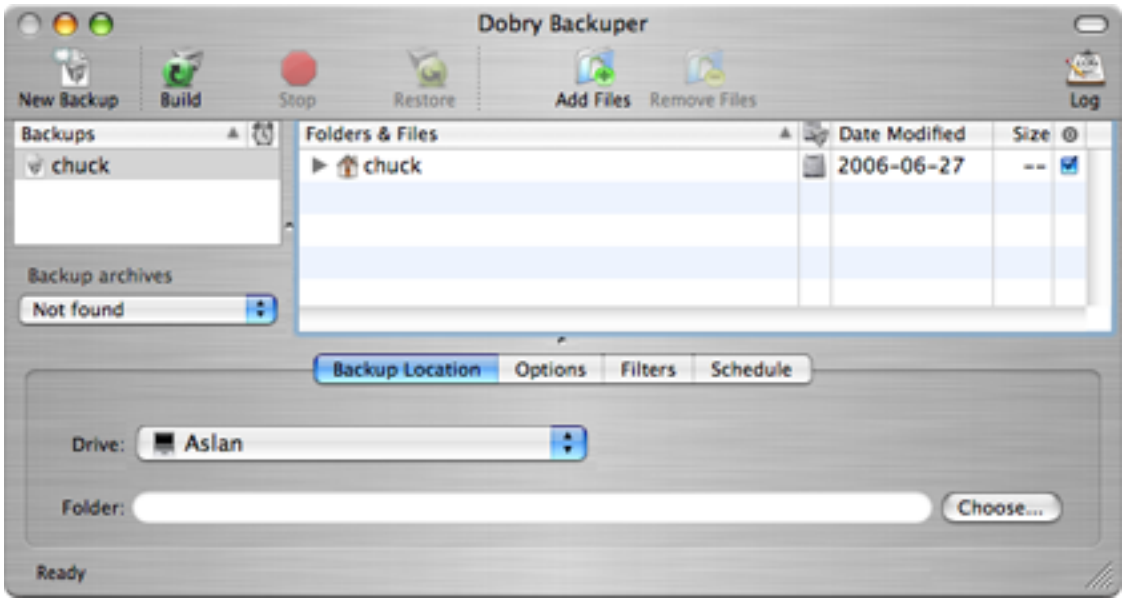
There is one, and only one, protection from data loss: backups. You know this, of course. But people don't back up. It's generally considered too much of a pain, until you find yourself enduring the pain of retrieving lost data. Overall, I'm pretty good about backups, and I've gotten better, because I have lost data. A few years ago, I don't even remember why or how, I lost my entire iPhoto library without a backup. I don't remember exactly why I didn't have a backup of it. I think at the time the iPhoto library wouldn't fit on the internal drive of my PowerBook, and so I stored it on an external FireWire drive. That FireWire drive was also where I backed up files, and it was that drive that went south. I didn't lose anything else because I had the originals on my PowerBook, but the iPhotos were lost and I had to pay a pretty penny to recover them using a data recovery service.

All this is a quick introduction as to why you need a backup plan in case you don't already have one. These days, .Mac members, such as myself, can use Apple's own Backup program, which can back up to hard disk, optical disk, or .Mac account. This is what I've been using for a while, and it works fine. Barring a .Mac account, you can do the old-fashioned backup plan of manually making copies of your files periodically, but the best backup plan is one you can forget about until you need it, and for that, you need a backup program that can automatically execute backups without manual intervention.

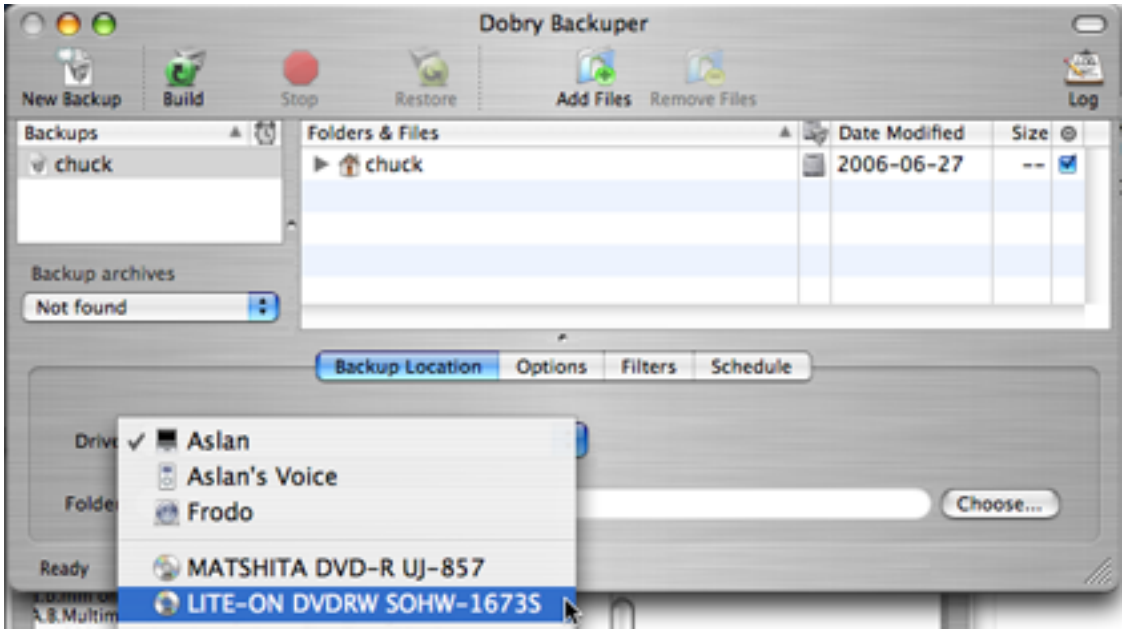
One such offering is Dobry Backuper, an inexpensive piece of software that will back up your files to the usual media such as CDs, DVDs, and external hard drives. One advantage

to Dobry's solution when compared to .Mac, is the use of compression, which for the data files that you're likely to want backed up, can save significant space. Another advantage is that the backup files created by Backuper are compressed with standard Unix compression tools, which means that even if you are unable to access the Backuper software, you can restore files using standard Unix tools or StuftIt Expander.

Using Backuper is very straightforward. Upon launching the program, you create a new backup by clicking the New Backup button, which prompts you to select the folder you wish to back up. I keep everything I can within my home folder (including my non-OS installed applications), so if I backup my home folder, I should get about 99% of what I want, the exceptions being those pieces of software that install something in the top-level Library folder, such as Saft, a program that augments the features of Safari for me.



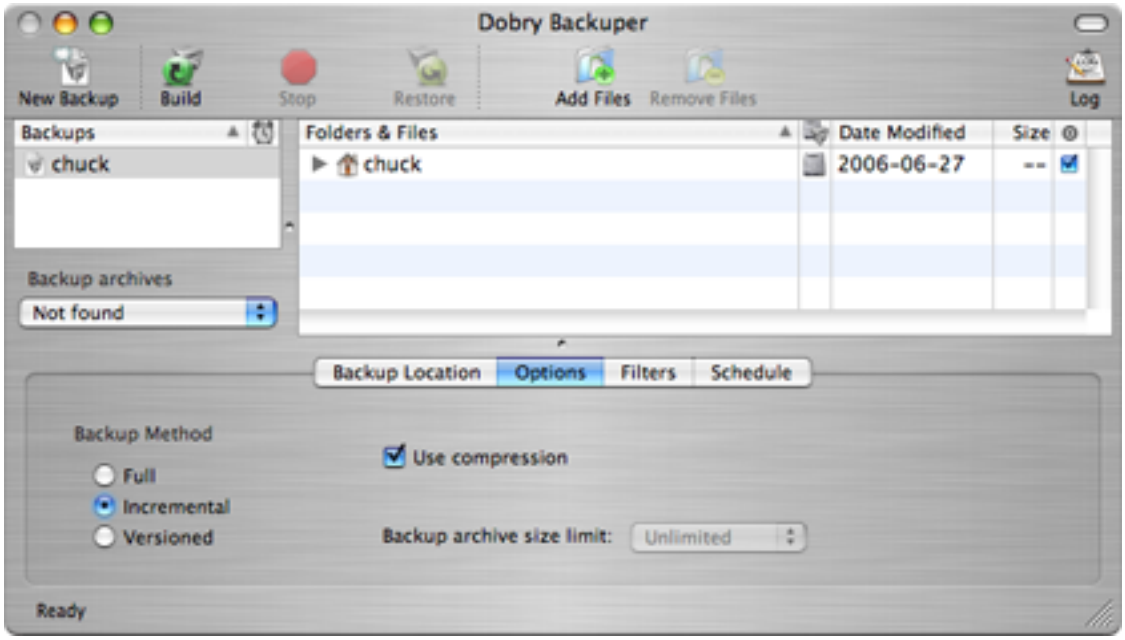
After selecting the files to be backed up, you select a location, such as a local or network drive, or an optical disk. If you're backing up to an optical disk, you'll be prompted to insert one during the backup. If backing up to a hard drive, you can select the folder into which the backup will be written.



After selecting the backup location, you can specify additional options. Backuper offers three types of backups: full, incremental, and versioned. Full backups mean that every time a backup is made, everything is backed up. Each new backup will remove the prior backup from the location on the hard drive. Incremental backups will only backup those files that have changed or been added since the last backup, and versioned backups will create full backups without removing the old backup. In most cases, incremental will be the best choice.

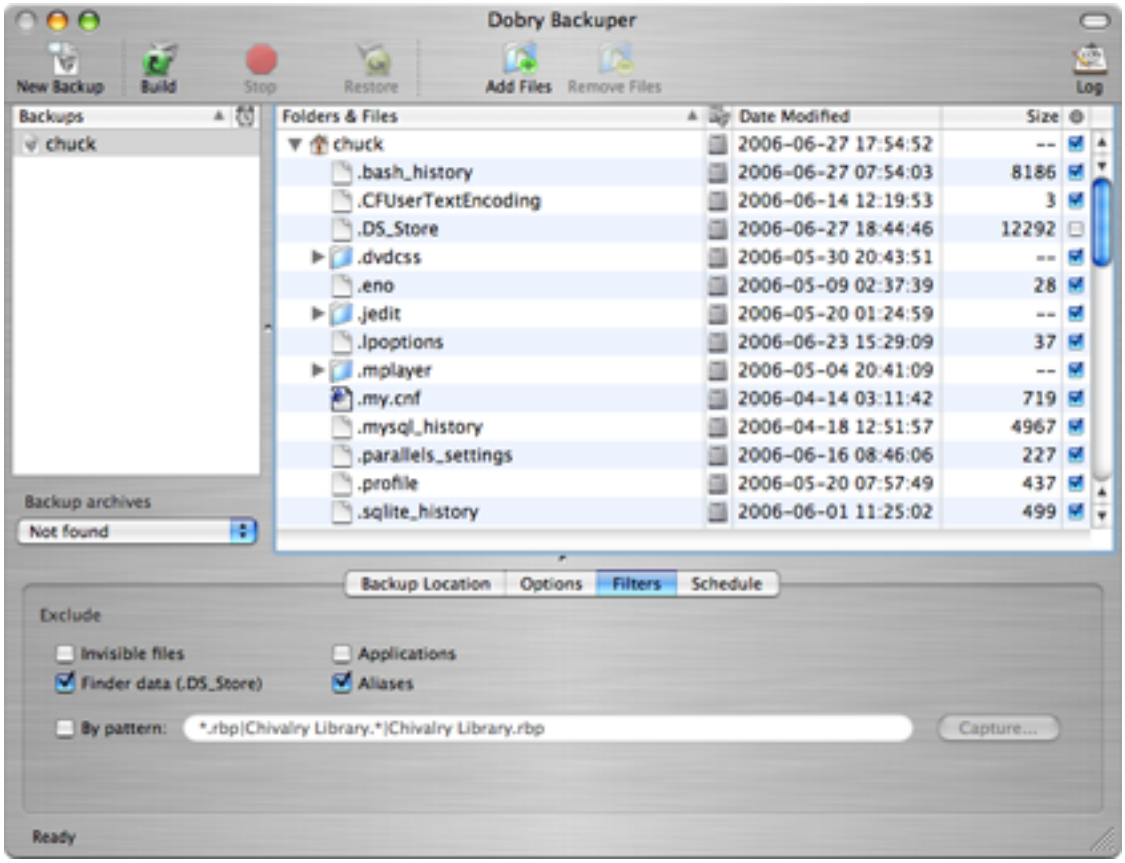
Another configurable option is whether or not to use compression and what the maximum archive size should be. This brings up a major disappointment for me with Backuper: for all intents and purposes, compression is mutually exclusive to backing up to optical disks. If you choose compression, the archive size limit selection is disabled. It is this option that allows you to back up files across optical disks. So an optical disk cannot be used with compression unless the entire compressed backup will fit on a single disk. My home folder is approaching 70 GB, which is never going to compress enough to fit on a 4.3 GB DVD.

Since backing up to optical drive entails writing a temp file to the system anyway, it would seem to me that it wouldn't be too much additional work to test the size of the archive after each file addition and, if it exceeds the optical drive capacity, remove that file and write the archive, begin a new compressed archive. This is, unfortunately, a deal breaker for me, and given this limitation, I wouldn't purchase the software.



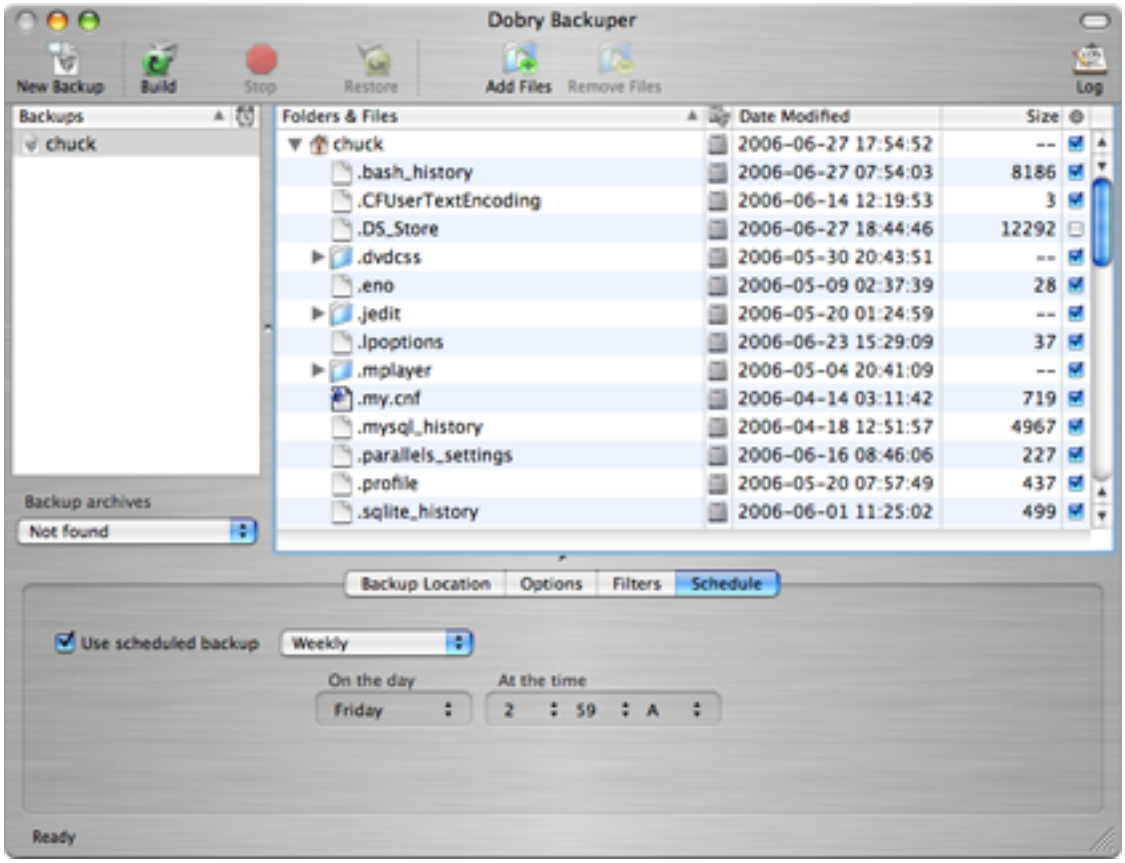
The other unfortunate feature lack is that the size of folders is not shown in the Backuper interface. In order to find out how large my home folder was, I had to Get Info on it in the Finder. This shouldn't be necessary, and I felt Backuper should be able to report the size of folders in addition to files.

But perhaps these not issues for you. Backing up to a hard drive with sufficient space allows compression. Moving through the features, you can also filter the files. Automatic filters include the exclusion of invisible files, .DS_Store files, applications, and aliases, but manual filters can also be used. Filters can be created for you by selecting a sample file that conforms to the pattern (such as choosing a file called MyFile.extention to exclude files that end in .extension). You can create more complex patters with a knowledge of regular expressions, which is a topic that non-programmers are unlikely to be familiar with.



Finally, you can schedule a backup to automatically occur at daily, weekly, or monthly intervals. The scheduling options are fairly simple, and backups fire as expected.

Even barring the issues mentioned above, in some ways, Backuper seems, well, unrefined. There are numerous interface glitches in the program that I found annoying. For instance, you can resize the space in a window allocated to the list of backups and files versus the backup options, but options will sometimes disappear, even when there is room for them in the tab section of the interface. I wasn't able to completely duplicate this bug and figure out under which conditions it appears, but I did notice it a couple of times. Also, the time options for scheduling a backup show only a single digit for the minutes, which looks strange, and there isn't enough space allocated to the AM/PM field to show "AM" when it is selected.



While it's difficult to recommend this software, it's also difficult to completely discount it. Backuper is early in its life (currently at version 1.5), and does work as advertised, even with the compression/optical drive issue and the interface issues brought up. The best I can say is, if you're looking for a backup program, and plan to back up to a hard drive, Backuper is at least worth a look at, especially given that the trial will work for 30 days, giving you plenty of time to evaluate if it will work for your needs.

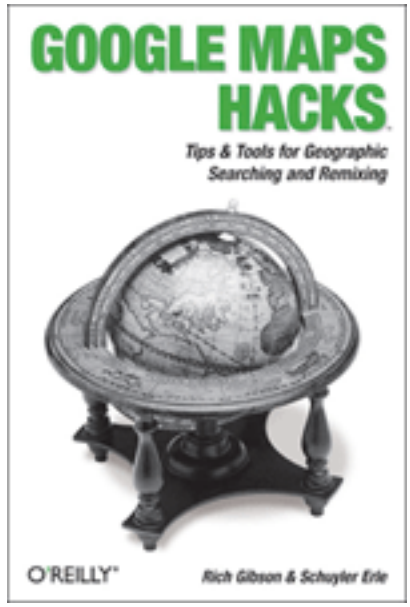
Book Review

by Wes Meltzer, wmeltzer@atpm.com

Google Maps Hacks

Publisher: O'Reilly
Authors: Rich Gibson, Schuyler Erle
Price: $30
Trial: Table of Contents, Index, Sample Hacks

The world is full of Google Maps mashups, and O'Reilly's Rich Gibson and Schuyler Erle set out to catalogue them, in the hope that they would be useful to their readers. When I think of these code-powered map tools, I think first of chicagocrime.org and Seattle Bus Monster, and of HousingMaps most of all. They're great tools, and well worth reading about.



Google Maps Hacks is just one volume in a series, ranging from *Skype Hacks* to *Google Hacks* to *Nokia Smartphone Hacks*, which outline a variety of interesting and innovative ways power users are using a particular tool.

I haven't read any of the other *Hacks* series books, but if they're anything like this one, they will not present themselves as especially useful. I say this as an extremely satisfied customer of O'Reilly's now-legendary programming references, which is what put the publishing house on the map in the first place. But I'm not sure who these *Hacks* books are aimed at, and

I am apparently not in this group. The subtitle on the book is dangerously misleading: "Tips & Tools for Geographic Searching and Remixing."

Originally I approached *Google Maps Hacks* as a way to implement something I've been trying to do for a long time, which is to convert a database of hotel information into a map with clickable tabs. I've never been happy with any of the existing solutions, either the code examples Google provides or mapbuilder.net, so I thought I would see what the pros had done, and maybe get some ideas about other neat ways to extend Google Maps.

While I found all kinds of interesting things, none of them had anything to do with what I wanted to do. My project must not be sophisticated enough to qualify. Some of the hacks presented—which, I feel I should note, are almost all real-world mashups by ordinary people—were genuinely awesome: traffic and cell phone reception maps, crime and news and weather maps, that kind of thing. They made me wish I was the kind of innovative thinker who could produce useful computer software, because some of them will change the world with their ideas. The map software is just the platform.

I would like to celebrate, then, the many amazing accomplishments of people much smarter than me, many of whom wrote their own chapters in the books. These are remarkably novel ideas, and go from the mundane, like sharing GPS tracking data using Google Maps, to the exotic things I mentioned previously.

But try though I could, re-reading the book three times over the last month while I hammered away at my problem, I found nothing in the book that would help me make my map work. All I wanted to know how to do was to design a map with markers in specific locations, labeled with the names of hotels, and make it so that a click brought up a useful information pane. I think it's fantastic that somebody figured out how to scrape Craigslist into Google Maps, but I'm not going to buy a book to find out about that. I'm glad I didn't purchase this book at the bookstore, because I would have been sad to return it, but it has a title which misleads you into believing that it will help you use Google Maps in your own site. It's certainly theoretically possible that this book could, but it did not help me.

The funny thing about the format of this book and the *Hacks* series, then, is that they seem designed more as an artist's showcase, provoking conversations about other possibilities, than as a collection of "Tips & Tools." The code's all present, of course, and it's possible that a more crafty JavaScript hacker might have been able to stitch several of these together into a seamless interface. But one of the big points of the O'Reilly books has always been that they are very nearly as usable by the novice as by the expert. This book did not seem to be.

I'm not really sure what to make of the experience. The "Tips" part of the book appears to be the hacks themselves, with introductions by the various programmers explaining how and why they did things in a particular way. The "Tools" part is presumably the first two and last two chapters, explaining some neat little tricks that you can get away with in making your own maps. While these tricks are indeed neat, they're very little, and the great bulk of the book is rather less useful.

Now my problem remains, and I have no new ideas about Google Maps mashups to share. Perhaps I'm wrong about *Google Maps Hacks*, and a thousand and one new mashups are forthcoming (just not from me). If that's the case, I will admit defeat and move along, and publicly apologize to O'Reilly for the poor review of this book.

But if it's anything like the "thought-provoking" conceptual art in certain galleries in the Art Institute of Chicago, I'm not holding my breath. Sorry, guys.

XIII

Developer: Feral Interactive
Price: $40 (from Mac Game Store, also part of the Big Metal Box 2 for the same price)
Requirements: Mac OS X 10.2, 800 MHz G3, 256 MB of RAM, 32 MB of VRAM, 1.6 GB disk space. Not Universal.
Recommended: Mac OS X 10.3, 1 GHz G3, 512 MB of RAM, 64 MB of VRAM, 2.2 GB disk space.
Trial: Feature-limited (two stages)

Based on a long-running French comic series of the same name, XIII features an amnesiac protagonist who may (or may not) have assassinated the American president. It lays a striking hand-drawn graphic style over the Unreal 2 game engine, giving it a distinctive and appealing feel. With cut scenes that play out an adapted story from the series, XIII has you traverse a dizzying series of locations in search of your past life.

Installation

XIII requires the DVD for installation and during gameplay (some of the movies play from the disc). Make sure to download the version 1.0.1 patch from Feral Interactive. According to Feral, it fixes several problems that could cause a crash mid-game.

Gameplay

Like many amnesia-stricken heroes, the game's main character (Agent XIII) struggles against both unrecognized adversaries and his own lost memories. The game borrows not only the cast of characters, but also a comic-art style, complete with sound effects. Walking feet and firing guns show up onscreen—very helpful when tracking enemy movement.

The "TAP" icon shows an enemy patrol in the next corridor.

The cartoon style gives the game a unique look and sets a good mood for the story. If only the rest were as creative! Players of other first-person shooters might find the objectives simple and the weaponry typical. You can strategize around the computer enemies fairly easily, as they often pay more attention to their downed comrades than to you.

Some enemies are well armored or move a little less predictably, but most of the game travels through straightforward levels with only occasional variety, such as using a grappling hook to reach higher and lower areas.

The grappling hook gets you through the electricity, if you time it right.

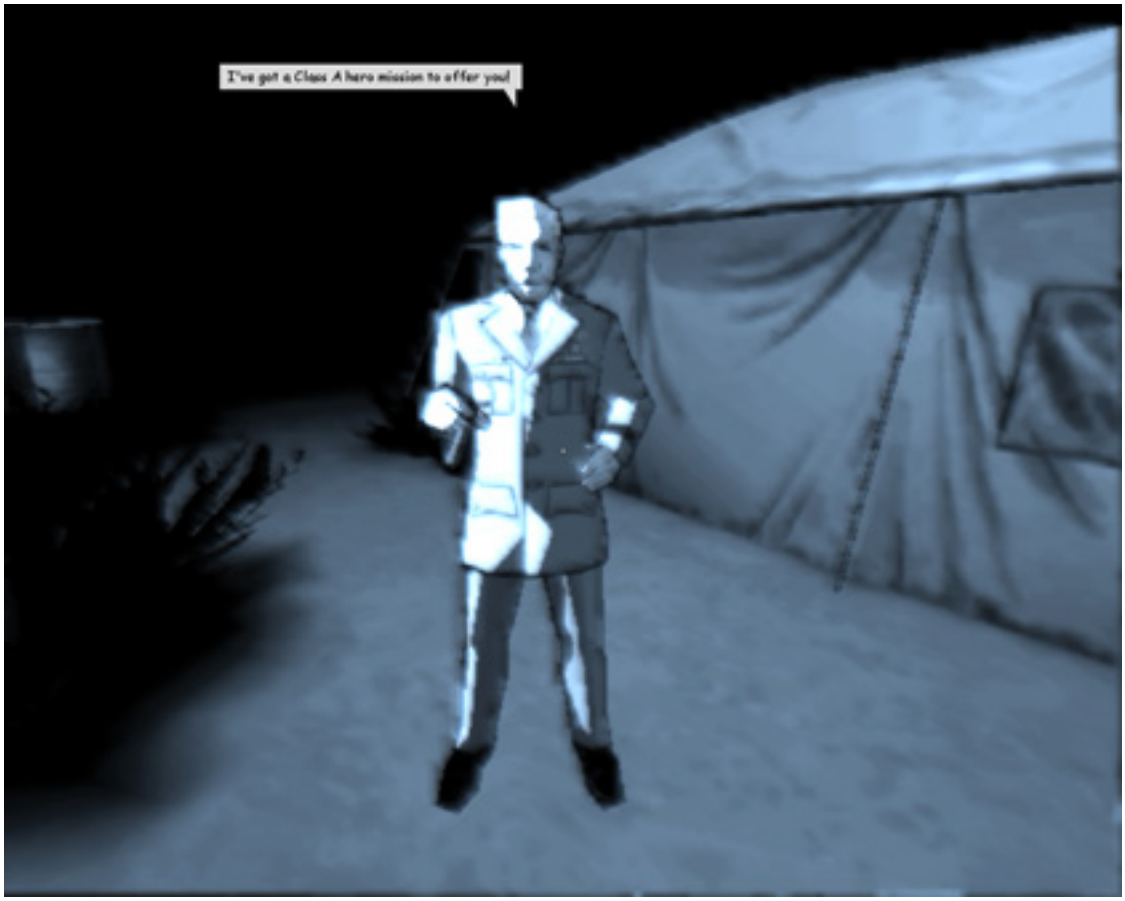
As you progress, long-range and stealth weapons tend to dominate. Go for the crossbow and sniper rifle when they're available. They allow you to target an enemy's head, which gives a one-hit kill and three-panel comic effect.

Crosshairs

In the story's opening chapters, you have to survive a series of violent encounters and escapes. You soon meet up with Major Jones, who apparently worked with you under General Carrington. The General now needs rescuing, and you're elected for the job.

Cell Rescue

Various story elements trigger memories for Agent XIII. You'll switch to a halo-like, black-and-white scene and briefly interact with one of the characters in the current storyline. These scenes usually raise more questions than they answer, which helps build the tension.



Flashback

It's worth noting that the game's *storytelling* is just fine. Even if the gameplay feels average, the writers did a good job of handling both plot and action. For some players—especially those who don't play the genre often—this may make the game worth it.

Throughout the game, major and minor goals guide you from place to place. They appear as text at the start of each mission, and you can push Esc at any time to review them on the pause screen. Usually, you find something, escape from an area, or protect a friendly character.

Major Jones

David Duchovny does the voice for Agent XIII, and Adam West (TV's *Batman*) does General Carrington. Perhaps I don't remember *The X Files* very well, but Duchovny sounds almost asleep through much of the game. West does a fine job as Carrington, though, so it's a wash for the two of them.

Annoyances

It takes a good while to unravel the mystery and finish the game. However, there's a cliffhanger of sorts (also taken from the comic series). Should XIII not spawn a sequel to "wrap things up," the story will simply be left hanging. For me, this counted as an annoyance, akin to waiting 20 years for the other three *Star Wars* movies.

The game did crash a few times, but it seemed to happen in only one particular area where the graphics card might have been at fault. Unfortunately, I didn't hear back from support requests to both Feral and the original developer, so I'm not sure if other people will have the same problem. (Eventually, I got past it without a crash.)

Multi-Player

XIII supports both LAN (local area network) and Internet play. Internet play goes through GameRanger, which is a free download from GameRanger.com.

To find other XIII players, create a (free) GameRanger account and enter the GameRanger lobby. The lobby lists all available games by name. Pick an existing game or host your own and wait for other players to join. (I didn't see any games hosted by others in the three or four times I looked for them, but you can probably arrange a time to meet with players in a Mac gaming forum.)

The familiar game types such as Death Match and Capture the Flag are there. One XIII-specific addition is "Sabotage," in which you plant a bomb at specific points in the level to gain points. Planting the bomb takes several seconds and leaves you exposed, so you probably need a few friends to protect you.

Sabotage: X marks the spot to plant the bomb.

Summary

XIII offers a unique visual environment and interesting storyline, but could use more variety in level design and basic gameplay. If you don't play first-person shooters very much,

this game will entertain but probably not pull you into the genre. For some players, the characters and story will be enough. For others, the ideas are familiar and executed more satisfyingly in other games.

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?

We currently need several **Contributing Editors**. Please contact us if you're interested.

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.