

ATPM

14.04 / April 2008

Volume 14, Number 4



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

Cover Art

Copyright © 2008 [Grant Osborne](#). We need new cover art each month. [Write](#) to us!

The ATPM Staff

Publisher/Editor	Michael Tsai
Managing Editor	Christopher Turner
Reviews Editor	Paul Fatula
Web Editor	Lee Bennett
Copy Editors	Chris Lawson Linus Ly Ellyn Ritterskamp Brooke Smith <i>Vacant</i>
Webmaster	Michael Tsai
Beta Testers	The Staff
Contributing Editors	Eric Blair Mike Chamberlain Chris Dudar Ed Eubanks, Jr. Matthew Glidden Ted Goranson Andrew Kator Robert Paul Leitao Wes Meltzer Sylvester Roque Charles Ross Mark Tennent Evan Trent <i>Vacant</i>

Artwork & Design

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

Emeritus

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

Contributors

Lee Bennett, Morgan Bowe, Tom Bridge, Ed Eubanks, Jr., Paul Fatula, Matt Johnson, Robert Paul Leitao, Isweeney, Wes Meltzer, Narcis Parfenti, Hector Ramos, Ellyn Ritter-skamp, Charles Ross, Mark Tennent, Ken Zindle, *Macintosh users like you.*

Subscriptions

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

Where to Find ATPM

Online and downloadable issues are available at the [ATPM Web Site](#). ATPM is a product of ATPM, Inc. © 1995-2008. All Rights Reserved. ISSN: 1093-2909.

Production Tools

Acorn, Apache, AppleScript, BBEdit, Cocoa, Docutils, DropDMG, FileMaker Pro, Git, GraphicConverter, L^AT_EX, make, Mailman, Mojo Mail, MySQL, optipng, Perl, Photoshop Elements, PyMesh, PyObjC, Python, rsync, Snapz Pro X, ssh.

Reprints

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

Legal Stuff

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



Thanks for reading ATPM.



Sponsors

About This Particular Macintosh has been free since 1995, and we intend to keep it that way. Our editors and staff are volunteers with *real* jobs who believe in the Macintosh way of computing. We don't make a profit, nor do we plan to. Our aim is to produce a fiercely independent magazine, with multiple formats designed for easy reading rather than showing ads. The views expressed in these pages have always been our own, and to help prove it we do not accept direct sponsorships or advertising. We do, however, need to pay for our Web site and other expenses, so we rely on minimal advertising, sold indirectly via Google and Yahoo, as well as the support of ATPM readers who shop using [our links](#).



Welcome

by Robert Paul Leitao, rleitao@atpm.com

Welcome to the April issue of *About This Particular Macintosh!* In this month's *Welcome*, we highlight the release of Apple TV, Take 2. This becomes unofficially our "Take 2" issue. Please take two issues of ATPM and pass one along to a friend or Mac-friendly family member. Our efficient digital distribution scheme saves trees and save you time in finding the best Apple news, views, and reviews prepared in our unique style and easy-to-read format.

Apple TV, Take 2

A couple of weeks ago, the writer of this column purchased an Apple TV, Take 2. Intrigued by the product description and owner reports, he plunked down the \$329 price of admission for enjoyment of the 160 GB model. On the way home from the Apple Store and with one of his teenagers in tow, he stopped at a local electronics store thinking he needed a new home theater system to accompany his new Apple TV. Arriving at home, it was the last he saw of the just purchased items.

One teenager gingerly carried the boxes in the house while another quickly found the power drill the writer had given up for lost a few months earlier. Cables were seen everywhere in the family TV room, and whirring sounds were heard outside the room as another teenager joined the installation crew. Three hours later, the same room was filled with the sounds of laughter. YouTube on a big screen was the cause of the merriment. Since installing the new Internet-connected music and movie playback device, the writer's home life has not been the same. Movies, music, and amateur video are streaming into his TV room and through the now all-important center channel almost non-stop. The writer reports his general happiness with the device: his kids have finally learned how to share a remote, he's canceling the family's NetFlix membership, and his teens are buying the content on their own iTunes accounts. Doing the finishing work on the holes made by teens with a power drill is his only remaining chore.

iPhone, Take 2

At press time, the Mac Web is abuzz with stories of Apple's plans for the manufacture and release of a 3G iPhone. Not satisfied with the success of the current version of Apple's combination of a video iPod and phone, it's as if life has come to a halt for thousands of obsessed fans pending the release of an update. Few times in recent history have so many people been interested in searching through otherwise obscure and arcane FCC filings for hints about approval of a product. Meanwhile, the millions who are already enjoying their iPhones view this preoccupation with amusement. It's not as if the current iPhone or the next iteration will be the last iPhone most of us will buy. Two years from purchase passes quickly no matter the iPhone in hand. It's waiting on the expiration of a non-iPhone contract that seems to take forever.

2 Take iPhones Too Many Times

Two Apple retail store employees from the Rockingham Park store in Salem, New Hampshire have been arrested on charges of stealing 332 Apple iPhones. This is grand theft by any measure and solves a smidgen of the continuing “missing iPhone” mystery.

2 Tunes Cents For Every Dollar Spent

The US dollar’s value has reached a nadir in relation to many world currencies. The devaluation of the greenback has caused an uptick in domestic prices and is listed as a primary concern among economists in the discussions and continuing debate about interest rates and economic policy. But there’s one New Economy goods trading instrument that is maintaining its popularity and value: the iTunes gift card. It’s available in a variety of denominations and at any number of local retail outlets.

With the Juniper Visa Card with iTunes Rewards program available through the Apple Web site, consumers can earn two cents toward iTunes gifts cards in \$25 denominations for every dollar spent through Apple’s own retail channels. The rebate is one cent per dollar spent on all other purchases. A note of caution: loading up the card with every monthly automatic payment to quench a thirst for iTunes currency while also using it for every day expenses can create a liquidity crisis of its own, as the writer of this column discovered when the fat monthly bills eventually arrived. Still, with a bag full of iTunes gift cards to accompany his new Apple TV, there’s no shortfall in entertainment content in his home. He’s holding some of his iTunes currency position in reserve for the iTunes iPhone software store for strategic gaming investments to put his iPhone’s accelerometer through its paces. It’s a form of currency that may hold its value and can be used as a hedge against the need to venture out to find music and movies at bricks-and-mortar retail stores.

AirPort Express, Take 2

Apple has put its AirPort Express wireless device on the fast track with an upgrade to the 802.11n standard. Able to handle up to ten simultaneous users, the AirPort Express adds AirTunes functionality to your wireless network, and it may also be useful in extending the range and strength of your wireless network from the AirPort hub to the location of your Apple’s TV. Upgrading the AirPort Express to the 802.11n standard may seem like a minor change, but for those buying or renting content via of iTunes, the upgrade brings all of Apple’s current wireless networking gear up to the faster 802.11n speed.

Blu-ray Dismay

Sony’s Blu-ray technology has won the HD content format war, but Apple appears to have a bystander in the outcome. Blu-ray drives are not currently available from Apple. As the premier manufacturer of personal computers for the content creation industries, it’s troubling to some that Apple hasn’t quickly added Blu-ray drives to its Macintosh computers even as a BTO option. We’re hopeful that the Blu-ray dismay will be resolved by this May and in time for reporting the news in our next issue.

Our April issue includes:

Bloggable: Shallow Depth of Field

Wes Meltzer considers the iPhone and the various discussions on what we should possibly expect at the SDK finish line. There's also plenty of opinions about background applications. This, and more, in *Bloggable*.

MacMuser: Chips With Everything

Whether it's food or software, Mark Tennent knows what he likes, and what he doesn't like.

MacMuser: Confused.con

Mark Tennent shares some punditry about DRM.

FileMaking: Getting Relational

This month, Charles Ross discusses relational databases.

Desktop Pictures: From ATPM Readers

Several ATPM readers provided this month's photos featuring a Caribbean sunset, Puerto Rico, Singapore, Indonesia, Romania, and more.

Cartoon: Cortland

Back in meatspace, Todd turns the tables on the enemy with a reprogrammed Lisa. Cortland and Angela arrive safely from the Mudrix, but Cortland has paid a heavy price.

Review: The Book of Wireless, 2nd Edition

Tom Bridge spends some time with a book intended to help you master the magic of Wi-Fi.

Review: Newer Technology iPhone Accessories Roundup

Newer Technology recently introduced a nice-looking array of iPhone accessory products. While one received the reviewer's accolades, the others fell somewhat short of expectations.

Review: PhotoAcute Studio 2.77

PhotoAcute Studio might be an invaluable tool for improving noise, geometry, and aberrations, but only under certain not-so-common circumstances.

Review: Take Control of Permissions in Leopard

"If your Mac has multiple users, sooner or later you'll need to know how permissions work."

Review: Wikipedia: The Missing Manual

Wikipedia: The Missing Manual fills in the gaps, just like you knew it would.



Xslimmer 1.5

Monolingual, an open source program, does this as well, and has for quite some time—for free. However, it doesn't offer the backup option or have a programs blacklist, so I only use it for removing unnecessary languages and input managers.

—*David Zatz*



Thank you for your informative review on Xslimmer. I saved closed to 5 GB of space when I was done. However, watch out for Adobe programs: I did strip the extra languages out from Adobe Acrobat Pro 7. It disabled the application! The most unpleasant part of it was not the CD re-install, but to get Adobe tech support to simply confirm the correct upgrading sequence to 7.x available from their Web site afterwards. Xslimmer is a great application indeed. And absolutely designate an external hard drive for the “back-up.” I learned the hard way and had to delete dozens of back-ups by hand!

—*Catherine von Dennefeld*

FileMaker 9

Good article, thanks. I upgraded to FileMaker 9.0 for one reason alone, and one reason you don't list here—a new function called “List.”

My use of this new function lets me create a list of strings based on related records. I had trouble doing this before, perhaps out of ignorance, but it is now very easy. This function is new to 9.0.

—*Jason Wohlstadter*

I didn't mention the List function because it's not new to FileMaker 9. It first appeared in version 8.5, and I covered it briefly in my [column](#) on that version.

You are, however, correct that the List function is useful for getting a list of related contents. Before its introduction, the only way to accomplish this was either with a script or with the use of a plug-in.

—*Charles Ross*

LicenseKeeper 1.3.2

For this purpose, I acquired some time ago [SecretBook](#), and I'm very satisfied with it.

—*Timotheus*



I've been using LicenseKeeper for eight months and have not suffered the difficulties that Eric has. As I recall, perhaps two or three times the serial number was not grabbed correctly from an Entourage e-mail import, but that was about it. I am grateful for the program's ability to import e-mails and the information therein. What a great feature in my opinion.

—*Phil*



Since I have an IMAP e-mail account, I just throw all my license e-mails into a folder that I created for that purpose. No special software required, and with Spotlight, it's a piece of cake to find the e-mail I need if I have to reinstall an application. If it's something I can't live without, I'll create a secure note in my keychain for the license info.

I must admit that I'm kind of puzzled at the proliferation of password, license, and other secure snippet keepers, given that Keychain Access is free, included with every Mac, and seems to provide decent security.

—*Brian Ogilvie*

The “Can Do, Just Works” Principle

Since you usually don't have to lug remote controls around, I prefer the big Panasonic to the too-tiny Apple. First, the Panasonic's buttons are labeled. Second, they are big. Third, the controls are positioned logically. Fourth, the power and mode change buttons are color coded. Apple's white-on-white, unlabeled design would be great for dealing with a clock radio or clock stereo at night: when working by feel, having just one tactile control area makes the remote easy to use. But, for controlling a complicated TV, surround sound, and video input set-up, a more powerful remote is needed.

—*Gregory Tetrault*

It all depends on the interface the Apple controller is using. Elgato's EyeTV is a classic case of a good interface, or, as is more usual, the Apple controller is using the interface built into the application running on the Mac.

In the case of the Panasonic, the controls are far from logical. Why, for example is the information button at the bottom of the number pad, when the control of the now/next menus is at the top of the controller?

—*Mark Tennent*



The problem with the Apple Remote is that it is too simple. It is fine for use with a Mac, but when packaged with my Apple TV, I need it to do a bit more. There is no volume control, so I need two remotes. I cannot change the TV input source from the cable to Apple TV. I cannot turn on the TV.

I would love to see a more advanced Harmony-style remote that is set up via USB.

—*Brad MacDonald*

It really does depend on the application running. QuickTime Player and EyeTV, the main reasons I use the Apple Remote controller, have volume control, fast forward/back, change channel, and so on.

I wouldn't expect the Apple Remote to control the more sophisticated functions of the TV, but then, once set they don't usually need tweaking again. It is a disappointment that Apple TV hasn't got more for the little Apple Remote to control. There are some excellent USB programmable remotes which might work.

I've not played with Apple TV because I find that an Elgato tuner with EyeTV 3 (or an earlier version with [CyTV](#)) is both cheaper and far more flexible than Apple's device and will stream live TV or recordings wirelessly without the need to prepare them especially for Apple TV.

—*Mark Tennent*

Outliners: A Progress Report

Good to see you back, Ted. Your comments, as usual, are astute and provocative.

On the one hand, I've moved to the Linux, open source world. On the other hand, in part because I'm no longer just starting out in the world, I regularly contribute financially to my Linux distribution of choice (PCLinuxOS) and bought a lifetime license to the most interesting Linux outliner I have found (Notecase Pro). I am willing to pay people to provide tools I like and use—and do not have the ability to produce myself.

Yet PCLOS is more a hobby (for texstar, the lead developer) than a business concern. And despite the remarkably swift progress of the Notecase Pro developer (mark and gather, hoisting, synchronizing of files, printing, etc.), I don't think the author is making a living on licenses, either.

So the question of business models is not trivial. Open source may mean that I'm a *little* less likely to be stranded with good but unsupported software. But things do move on quickly, as we all pursue maximum gadgetude. That makes it tough for innovators to stay in the game long term.

I look forward to your next piece.

—*James LaRue*

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 and may be edited for publication.



Shallow Depth of Field

Sometimes, my friends, fate smiles upon you.

You see, often it's hard for me to write about topics that are of breaking concern in the Mac community because I have a particular interest in the outcome. I bend over backward to make absolutely certain that I present both sides of the story fairly, but that doesn't make it easy.

Then, there's writing about the iPhone. You see, I don't have one, because I don't have AT&T service. And until my friends who do aren't constantly cursing dropped calls, I have no intention of switching carriers and buying an iPhone. My contract's come up for renewal, come and gone, and I just couldn't pull the switch.

So it's been a pleasure to sit on the sideline and watch the debate over user-installable iPhone applications. The object of my current cellular misery is a T-Mobile SDA with Windows Mobile, which you *can* technically install your own applications on. Let's just say I don't think that feature has improved my phone, your choice of software being generally limited to "programs that are not useful but are well-written" and "programs that are not useful and are poorly written." I have yet to find one that I would keep.

I'm curious to see exactly what user-installable iPhone applications will provide over the Web interfaces that are currently available. Apple announced the iPhone software development kit on March 6, along with an announcement of [enterprise-friendly features](#), and so far I've heard much discussion of the limitations and freedoms of the SDK but not much by way of the sorts of applications that might be forthcoming. *Macworld* published a [wishlist](#), and so did [Brent Simmons](#) but that's about all.

Anyway, let's talk about what *is* out there this month.

Jesper, of Waffle, came up with a fantastic [scorecard](#) for the announcement (live-blogged very well by [Macworld](#)). The verdict was that it came out better than he expected, but I'm not sure how to explain his methodology concisely. Just read it; it'll make more sense than if I tried.

But Jesper's excitement at the technical specifics is vastly eclipsed by 37signals' Jason Fried's enthusiasm. Fried writes, "What we saw today was the beginning of two-decades of [mobile domination](#) by Apple. What Microsoft and Windows was to the desktop, Apple and Touch will be to mobile." The key, if you ask Fried, is that the existing platforms (Palm, Windows Mobile, Blackberry, Symbian) are in many ways lackluster and lack traction. If Apple can capitalize on the success in a way that they couldn't in the PC arena, Fried

argues, it may be a long time before anything comes along that can displace the iPhone's success.

He's seconded by a whole host of commenters who love the idea behind the iPhone application store: [Craig Hockenberry](#), for instance, observes that Apple's 30 percent cut on the price of the software is fair for their handling of downloading, payment processing, and some of the promotional footwork:

Those things pale in comparison to the value of being associated with the Apple brand. Having their explicit stamp of approval and being included in the App Store will make any product more appealing to a customer. Buying directly from Apple means that your software won't screw up their phone and that can be returned if it doesn't live up to expectations. That, combined with the ease of a single click purchase, is going to drive a lot of sales. You'll make up that 30% without even trying.

Jens Alfke sees, in the potential of an application sold for 99 cents, the ultimate [impulse buy](#), like a pack of gum at the grocery store checkout line. And Panic's Steven Frank thinks the company's 30 percent fee pays for the [best promotion](#) money can buy: Apple's.

Out of Focus

The real bone most developers had to pick was on a specific restriction imposed by the iPhone SDK: third-party applications can't run in the background.

John Gruber had a nice overview of what this means, and how the iPhone handles background applications currently. He writes:

The iPhone is severely resource constrained. Battery, RAM, and CPU cycles are all severely limited. If third-party apps could run in the background, all three could suffer. RAM would suffer for sure; all running apps consume memory. The iPhone has just 128 MB of RAM, and no swap space. CPU performance and battery life would suffer when background apps do something—and if they're not doing anything, [what's the point](#) of keeping them running?

He followed that up with a couple of posts about the [other side](#) of [the argument](#). But even then, he asks, "How are typical users—not Ian Betteridge, not me, and probably not you, but typical users—supposed to know which apps are causing the problem?"

Craig Hockenberry returns with a nice post on background applications. His early versions of Twiterrific for the iPhone polled the Twitter server every five minutes, but that had a significant impact on battery life, so he scaled back his ambitions. After all, he asks, what happens if you have more than one application polling regularly at different intervals? He tries to imagine a [notification-based system](#) for polling, where the OS could send all

running background applications an alert that it was opening up a connection—and then says, “Do I expect such a sophisticated system to be available in a beta of version 1.0? Hell no. And neither should you.” (And he has an interesting argument to be made, [later](#), that the limited real estate on a phone requires a completely different set of notification models for background applications.)

But Jens Alfke (seeing a theme?) begs to disagree. He thinks that it must be possible, since other phone systems use IM application services, for instance, and that requires a [persistent connection](#). These persistent connections are more efficient than polling, and he wonders to what degree it’s possible to work around the technical limitations. (I’d like to note that at least on my phone, the IM client appears to piggy-back off of the SMS notification service and connect to the IM server using a push-based proxy.)

Rogue Amoeba’s Paul Kafasis has a laundry list of things he wants from the iPhone SDK, which he’s published as a [list of bugs](#) filed with Apple. He, too, hooks on to background applications. I wonder what it is about that that’s captured developers’ imagination.

Just Plain Bad Vision

- Sven-S. Porst has a great round-up on how [localized application names](#) work. Apparently, there are some weird flaws having to do with inconsistencies in documentation, shortcut keys, and translation issues. Anyway, it was an enjoyable read. He also wrote a great explanation on how [Time Machine works](#), which is also pretty cool.
- The next version of Internet Explorer will be [standards-compliant](#). Gasp: imagine that. It only took Microsoft how many years?
- *New York Times* tech-blogger David Pogue got his hands on some hilarious [tech-support transcripts](#). “Listen. I’m pressing Control, eh? And nothing’s happening, eh?” Oh, my goodness.
- Charlie Rose, PBS’ newscaster, got himself injured rescuing a MacBook Air that he believed was in imminent danger. Let’s just recap: Charlie Rose, he of the immaculate personal appearance on broadcast television, chose to [rescue a new computer](#) over protecting his own appearance. Wow.
- M.B. Darden went to an Apple Store for the first time, apparently, in a while. He discovered that they’re quite the happening place at malls, even when they aren’t busy, and that [the PC war is over](#) and Apple has won. Isn’t *that* interesting? You’ll enjoy the rest of it, too.

And that’s all for March, folks. Come back next month, and maybe I’ll finally have gotten around to buying a phone that doesn’t run on Windows. If nothing else, there’s always some Mac news to write about, hmm?



Chips With Everything

What will you want for dinner, or shall we eat out? A few weeks ago there was the tussle with “Can I have your birthday list?” This resulted in a virtually blank piece of paper. After all, an Aston Martin DB9 is no more likely than being stuck on a desert island with a broody Charlize Theron.

Most of the meals we’ve had out have been pretty dreadful, too. Expensive, badly cooked food with wine unfit for vinegar. The worst being in a Brighton Tex-Mex restaurant, which had a Les Routiers sign above the door. Only after reluctantly paying the bill we learned that the sign was from a restaurant the chef ran in another town some years before. It is just so frustrating when 50 miles away in France just about any restaurant’s 14-euro menu will be scrummy and even the cheapest plonk drinkable. We even tried one of Gordon Ramsey’s nightmare kitchens in Hampshire. Unfortunately, this was *before* rather than *after* the great man gave it an effing seeing-to. That one burned down in mysterious circumstances not long after the TV show was broadcast.

The answer can only be cod and chips from the Fish Hut on the beach in old town Hastings. Bought where the cook’s apron is as greasy as the spoons, no Les Routier sign, but a far more valuable certificate from Hastings Beaver Scouts have voted the Fish Hut the best chippie in town. The portions are far too large and eaten while walking between the fishing fleet pulled up on the beach and throwing chips to flocks of gulls hovering above us. Especially baby ones with spotty grey feathers who haven’t learned the tricks of begging.

The only other meal worth repeating is from the Still and West, sitting in the window overlooking Portsmouth harbour. It’s a bit like being in the stern cabin of *HMS Victory*. Her Majesty’s ships pass about 20 feet away, their crews lined up on deck to enter harbour at eye level with you as they cruise by. Fish platter for two with chilled Muscadet sur lie, please.

What of Software?

Ask a similar question about software and the answer comes easily. What is the best you have used? Both of us said QuarkXPress without thinking. We’ve lived with it since version 3, though we nearly switched full-time when InDesign 2 arrived while QXP 5 was still running in Classic. The latest, QXP 7, has leapt ahead in terms of usability and speed. And it has an interface based on a long-established design. We were loud and vehement in our slating of QXP 6, but even InDesign CS3 is flawed in so many departments that it’s going to take Adobe a long time to dig itself out of some of the holes.

Other packages have transformed our work. PostScript, PDFs, Photoshop, and Acrobat have been as life changing as XPress, but not without side effects. Plus, of course, Mac OS X—especially Quick Look and Time Machine, iTunes for revolutionizing music, and Call of Duty United Offensive online and EyeTV with CyTV for the fun they give. iDisk is so darned useful being linked closely to the Mac, and iChat is a brilliant idea waiting for the world to catch up. Some programs have been around as long as we've been using Macs, such as Fetch, the most reliable file transfer application, even if we prefer Transmit because of the support for WebDAV and columns view.

And Now the Baddies

We hate Microsoft Office with a vengeance. In fact, we hate everything MS because it sells products that seem to have been made by 1970s British Leyland and updated by local government ever since. MS can even buy-in a great piece of work such as iView and turn it into the stinking pile now called Expression Media. Adobe products aren't far behind. After using most for the last 30 years, we still fight with them and will never forgive Adobe for buying and dropping Freehand—which was always a better package than Illustrator.

Second worst to those. . .hmm.

Copyright © 2008 Mark Tennent, mtennent@atpm.com.



Confused.com

This DRM thing is starting to take some funny turns.

From next year, the UK government proposes to make it [an obligation of ISPs](#) to stop illegal file sharing in an attempt to combat piracy. Meanwhile, in America, a [recent court ruling](#) has put the knockers on the music industry's fast-path copyright infringement cases against file sharers. It will not be enough for the RIAA to show that copyrights have been infringed by the contents of someone's computer disks. They will have to prove the material was actually being distributed.

Just to muddy the waters even further, the Electronic Frontier Foundation has discovered that every track downloaded from iTunes [contains](#), hidden inside, the name and e-mail address of the person who downloaded it.

Then Martin Lewis' Money Saving Expert has put [DVD unlocking](#) into the hands of the average punter. It's legal, and the more savvy of us knew about it years ago, but now everyone will be able to do it. The whole point of DVD locking was for the film studios to control distribution of their films, charging different prices around the world, and to stop the importing of films from cheaper regions. One stupid element is that while it is legal to import the films, it isn't to buy them in the UK unless the British Board of Film Censors has approved them.

“Completely Ridiculous”

The government's ideas of how to stop illegal file sharing appear to be completely ridiculous. ISPs will be able to see and block peer-to-peer connections, but how on earth are they meant to inspect every bit of data running through their wires? They are no more able to do this than a car manufacturer is of stopping people speeding. Besides, if peer-to-peer sharing is stopped, there are plenty more ways to share files, and with the rise of faster ADSL and cable connections, the need for bandwidth sharing peer-to-peer is far less.

Martin Pitt, co-owner of Aquiss, an Entanet reseller and my current ISP, explains that the bulk of off-peak traffic is file-sharers. The service deteriorates noticeably in the evening compared with daytime use when we often have nearly the whole bandwidth to ourselves. This is frustrating because, as digital designers whose whole work is distributed to our clients electronically, we often need to FTP files back and forth all night. As these contain copyrighted material, how will an ISP know which is legal or not?

Phorm Filling

Just to cloud the already murky depths, Martin also [told me](#) about a company called Phorm that has apparently been working with companies such as BT, Virgin Media, and TalkTalk

to spy on users and log their browsing habits for the last year. This is so that advertisers can target their adverts more effectively.

All this DRM, spying, and suchlike is all very interesting, but it still doesn't tell me how to get my protected iTunes tracks onto an MP3 CD that plays in my car. Apple has [temporarily stopped](#) the [Hymn Project](#), a DRM stripper resource. So until a judge decides otherwise, it's a case of copying the tracks to audio CD then reimporting back into iTunes as MP3s before copying back onto an MP3 CD.

What a chore when I have legally purchased the tracks and don't want to lug an iPod around all the time. Especially as it can't be operated by the sound system controls on my steering wheel. As for my details being stored inside iTunes downloads, who cares?

Copyright © 2008 Mark Tennent, mtennent@atpm.com.



Getting Relational

[FileMaker](#) is described as a relational database. Most of us have at least an intuitive idea of what a database is. While we covered the relational aspect in some detail in the [first column](#), we're going to look more closely at its meaning here. Some of what will be presented here is theoretical, but it will have practical applications, and the only portions of relational theory that we'll cover will be those directly applicable to FileMaker databases.

What Is a Relational Database?

Long ago in the neolithic era of computing (1969), [Edgar Codd](#) wrote an [article](#) describing the [relational model](#) of database structure that is the basis of modern relational databases. Most modern professional grade database managers, such as [MySQL](#) and FileMaker, have their roots in this paper and later works by Mr. Codd. The word [relation](#) originates from the mathematical concept of a table, although one could also think of relation as stemming from the idea that the data is built around relationships between things.

The basic idea of Codd's relational model was that data should be broken into tables, and tables are made up of columns and rows. In FileMaker parlance, rows are records and columns are fields within those records. Tables and fields are chosen based on a process of [normalization](#), which is designed to ensure that the database is properly broken into tables based on the elimination of duplicate data, and to ensure data integrity.

If, after reading this article, you're interested in delving further into the details of relational database design and normalization, I can recommend a book that helped me quite a bit some years ago, [Relational Database Design Clearly Explained](#) by Jan L. Harrington. The link will take you to Amazon's page for the edition I'm able to recommend, as I'm unfamiliar with the [second edition](#).

Normalization

Normalization is a rather theoretical and complex topic, but in regard to building FileMaker databases, it can be broken down into the following guidelines:

- Each record in a table should be uniquely identifiable.
- Each field in a record should be dependent on only the record in which it exists.
- Each field in a record should be uniquely determined by the record it exists in.
- Each field in a record should store a single piece of information.

- Each field in a record should appear only once.

I'm playing loose with normalization terminology and definitions, but this is the basic idea behind normalizing your data to what's called the [fourth normalization form](#). By following these rules, you help to ensure that your database is structured so that data isn't repeated and when an attribute of one record changes, it's properly shown in all relevant areas of the system.

Example Design

For the most part, you can design a database without regard to what you will be using to build the database. For the moment, that's what we're going to do. After we've designed the database, we'll create our design in FileMaker.

Our example database will store a collection of books. Information we want to track about each book will include the title, author, publisher, location, and other items. We'll begin with a single table and create new tables as we apply the rules above. Some portions of the structure of our database may seem contrived, but they will serve to illustrate general principles.

Obviously, the first table we'll need is Books. The Books table will begin with the following fields:

Books
Title
ISBN
Author
Publisher
Publication Year
Location
Genre

Let's apply the first rule above. Are records within this table uniquely identifiable? Another way to ask this question is, does the possibility of duplicate records exist? As far as the database is right now, yes. It's conceivable that we could have two copies of a book stored in the same location, and we would have no way to tell within the database which book we're looking at.

The most common solution to this is to add a field whose sole purpose is to uniquely identify a record. This is sometimes called a primary key field or a serial number. In this case, we'll add a field called Book ID, which we will define to have a unique value for each record. FileMaker has mechanisms for ensuring that this is the case, as we'll see when we implement our database.

There are other solutions to the problem besides using a dedicated key field. One that is commonly used is to find a field that you're already using that uniquely identifies the record, such as a product number or perhaps a combination of fields that uniquely identify a record, such as the person's name (calculated from a first, middle, and last name). However, I nearly always opt for a key field that serves no purpose other than identifying a record. There are a number of reasons for this, but I think the most important one is that it's very useful to have the key field be not only unique, but static. A person's name may change, while in order for relationships between tables to have integrity, key fields should not. Additionally, unless there's overwhelming need, I do not display the key field to the end user, using it for the background programming I build instead.

Given what we've done to the Books table to conform to the first rule, our table now looks like this (Book ID being underlined to indicate that it is the primary key field):

Books
<u>Book ID</u>
Title
ISBN
Author
Publisher
Publication Year
Location
Genre

Let's move on to the second rule. Is each attribute we are tracking dependent only upon the record in which it exists? No, because the author, publisher, title and publication year all depend on the ISBN number. If we have two copies of a book, we'll be storing this data twice and if it's been entered incorrectly in both records, updating one of them will not update the data in the other record.

Solving this problem will require the addition of another table, which we'll call Titles. We'll move the ISBN, Title, Author, Publisher, and Publication Year fields from Books to Titles, and add a new field to the Books table to indicate which title a book belongs to. We'll also add a Title ID field to the Titles table as we did with the Books table. As a general rule, always include a primary key field in each of your tables.

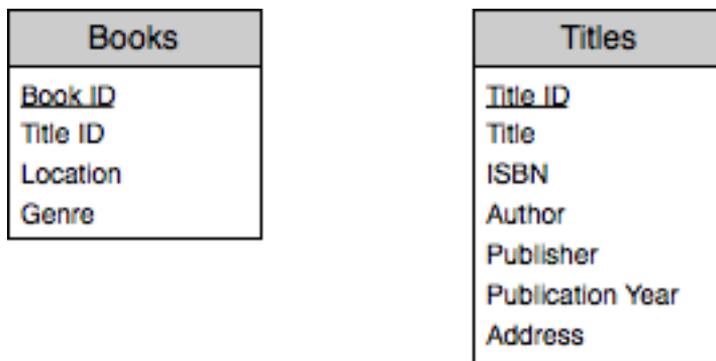
Books
<u>Book ID</u>
Title ID
Location
Genre

Titles
<u>Title ID</u>
Title
ISBN
Author
Publisher
Publication Year

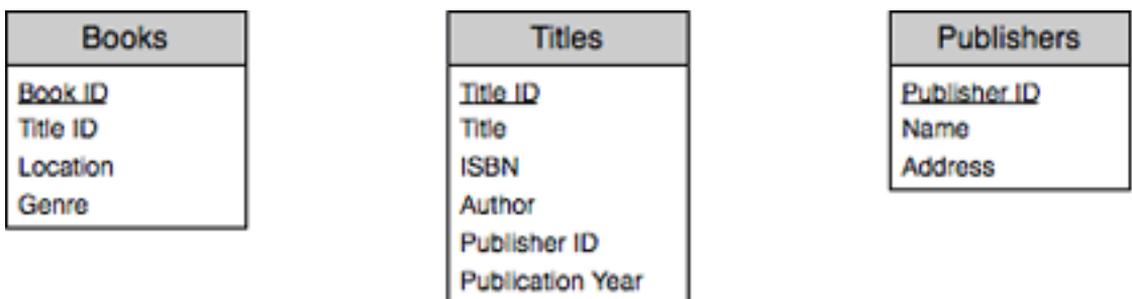
If you're thinking that the ISBN number could have been used as the primary key, you're correct. It satisfies the basic requirements of a primary key field in that it should be unique, given a specific title, and therefore will uniquely identify a record. A valid reason for a dedicated primary key field in this case is that if we use the ISBN number and then have books that reference the ISBN number, but someone incorrectly entered and then corrected the ISBN number, the link between the book and the title would be broken (as you'll see later when we build the relationships in FileMaker).

As our database stands now, it already satisfies the third rule, so let's add a feature, first in a manner that will break the third rule, and then correcting the structure to conform to it.

Perhaps we'd like to track further information about the publisher, such as the publisher's address. Adding this information to our Titles table alters the structure to appear as follows:

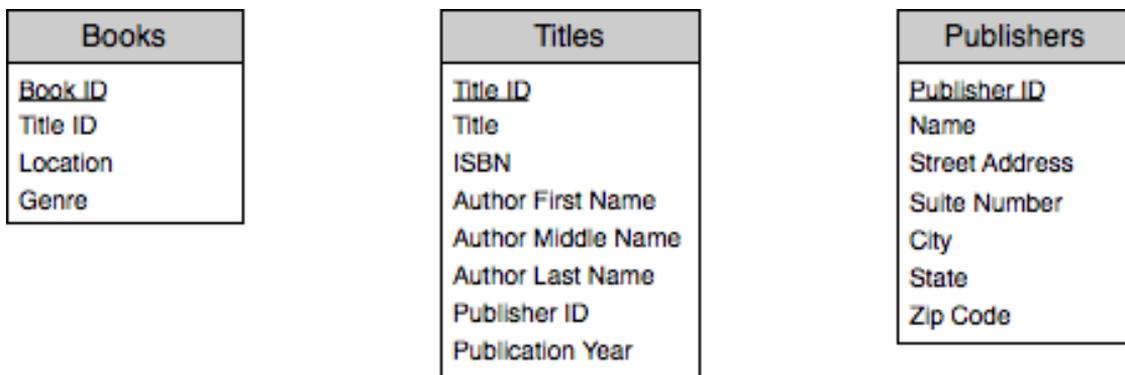


Now we have the ability to track the address of a publisher as well as the other information about a title. The problem with this is that it breaks the third rule. The third rule says that a field in a record should be uniquely determined for that record, and this isn't the case with the address. The title uniquely determines the publisher, and it is the publisher that uniquely determines the address. Therefore, we need another table, this time for Publishers, and move the publisher name and address fields from Titles to Publishers, adding a Publisher ID field to Titles so that we can track which publisher is associated with a title.

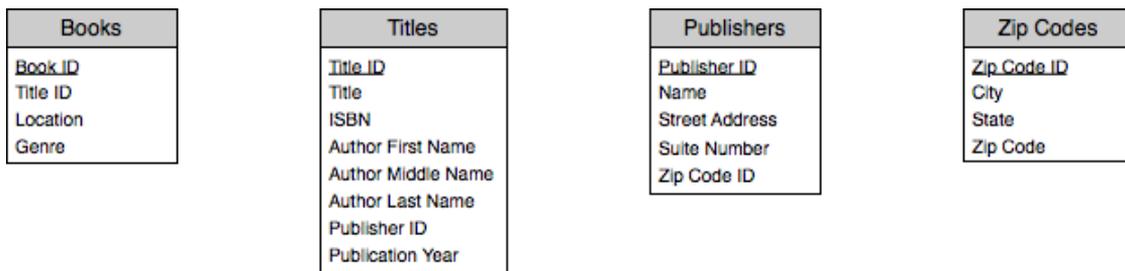


At first glance, our database seems to satisfy the fourth rule, but closer inspection shows otherwise. For example, the name of an author is actually made up of the first name, middle name, and last name, three separate pieces of information. Also, the address field in Publishers is probably made up of a street address, possibly a suite number, the city, state, and zip code (we'll limit ourselves to US addresses for simplicity). To correct this, we break up both of these fields into their respective components.

While it may seem overkill to break a single name into multiple components, as in the case with the name and address, it will make future development easier. For example, without breaking the name into its components, it would be difficult to sort the titles by author last name, and leaving the entire publisher address in a single field would make it difficult to search for all the publishers within a specific state.



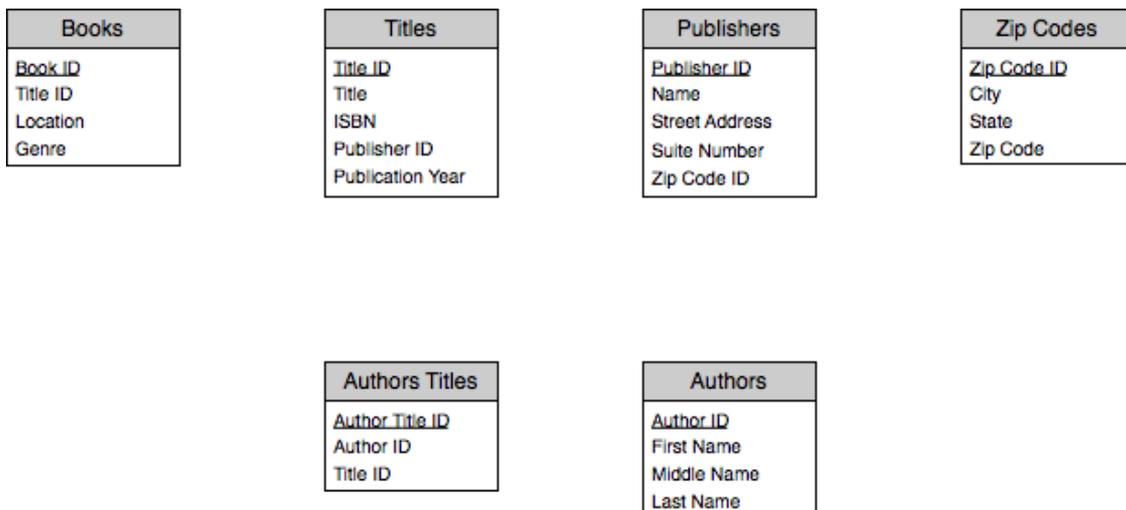
With these changes, the Publishers table now violates the third rule! A publisher uniquely determines a zip code, but the zip code will determine the city and state. Proper normalization will require that we create a zip codes table to correct this.



Our final rule, that each attribute appear in a record only once, initially seems satisfied as the database currently stands, until we consider that a book could be written by multiple authors. A common way for novices to solve this is to place multiple versions of a field in a database, so that we would have fields for Author 1 First Name, Author 2 First Name, etc. This presents two problems. First of all, we limit ourselves to the number of authors that can be assigned to a title. If we add three sets of author name fields and come across a title with four authors, we have to edit the database design, and again when we come across

one with five authors. But we also have the problem that when we have many titles by the same author, we're entering that author's name many times throughout the database.

Therefore, we should break the authors out into its own table. But we need to go a bit further than this. Not only may a single title be written by multiple authors, but a single author may write multiple titles. All the relationships spoken of thus far have been one-to-many (a single title can have many books, a single publisher can have many titles, etc.). In this case, we're working with a many-to-many relationship, and that requires an extra table, often called a join table. So we end up not only adding an Authors table but an Authors Titles table to track which authors have worked on which books. Our structure becomes as follows:



Breaking the Rules

We now have our tables designed to conform to all of the rules outlined above. At this point, I'll mention that the rules are not set in stone. There are times when you will break the rules for some specific purpose, but you should know the rules and why you're breaking them before doing so.

As an example, perhaps the overhead of dealing with a Zip Codes table is more trouble than you think is necessary. In that case, you might decide to keep the city, state and zip code fields in the Publishers table instead. Doing so is fine, and FileMaker will allow it. The point is, know why you're doing so and be aware of the potential drawbacks (duplicate data entry and therefore a possible lack of data integrity).

Another example, again from the Zip Codes table, is the primary key field we used. In this case, you might not want to use a dedicated primary key field, as you may decide that the Zip Code field itself will adequately serve as a primary key. Or, perhaps you note that a combination of Author ID and Title ID in the Author Title table can serve as a primary key, and remove the Author Title ID field. Again, doing so "breaks" the rules, but for an arguable reason that may be quite valid.

Normalization Complexity

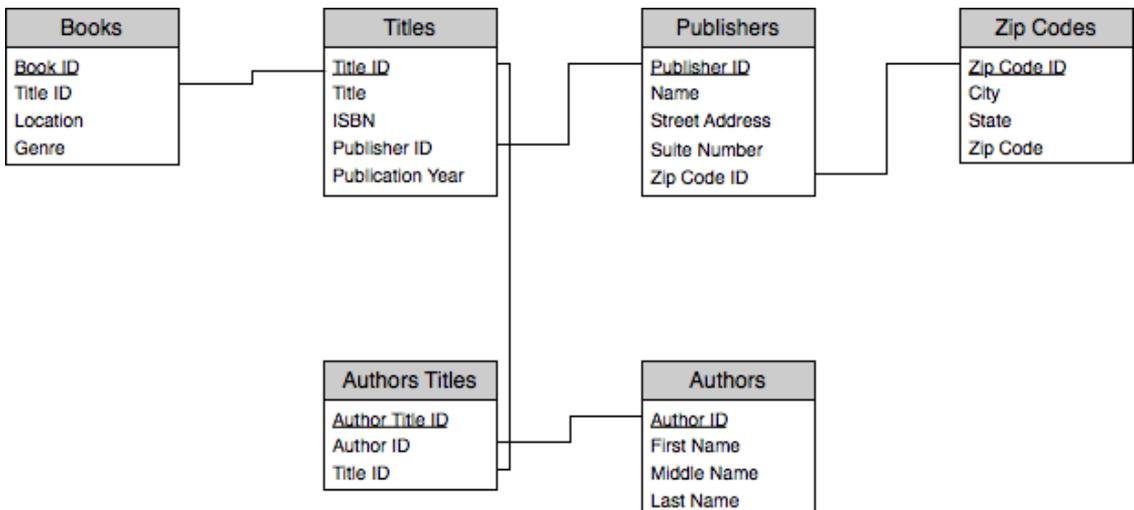
While it may seem at first that applying the rules of normalization is time-consuming and tedious, it eventually becomes second nature. Honestly, I haven't thought about the precise normal forms for quite some time. After you've applied the normalization rules to a number of database systems, you'll be able to create a database design that is normalized without having to think about it. It will simply be obvious that, for instance, Authors should be its own table and will require a join table to properly store how authors and titles are related.

Designing Relationships

Now that we have our tables and have confirmed that they conform to the normalization rules, we can begin to design the relationships between them. This will allow us to make use of each table's data within other tables. For example, although we have a Books table, the title of the book is actually stored within the Titles table. We have a Title ID field in the Books table to enable us to link the tables together so that when viewing a book record in FileMaker, we'll be able to see the title of the book, even though it's not actually stored in the Books table.

The design of relationships is often best done graphically, using boxes to represent tables and lines to demonstrate the links between tables. I use [OmniGraffle](#) for this, as it has built-in arrowheads to serve this purpose, but any program that allows simple drawing will do, even FileMaker itself, using its built-in layout tools. And, of course, there's always the low-tech paper-and-pencil solution.

First, let's show the diagram, which is called an [Entity Relationship Diagram](#), or ERD, for the tables we've designed.



What we're showing here are the basic relationships between each of our tables. For example, you can see that the Books table is related to the Titles table based on the Title ID field in both tables. This means that to link a title to a book, we find the Title ID that

uniquely identifies the title we are interested in, and assign this value to the Title ID field in the Books table.

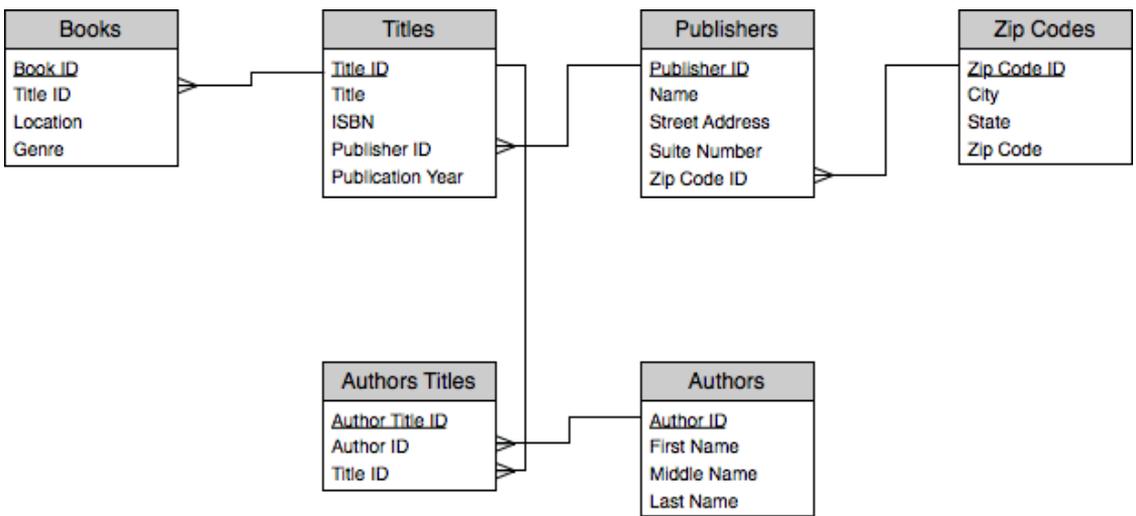
A further step in database design is to use the lines connecting tables to indicate not only that there is a link between the tables, but what sort of link it is. Links can be described in at least two ways (there are more ways to describe links which we won't get into yet). The first way is how many records in one table link to how many records in another table. As you might guess, this breaks down to three possibilities: one-to-one, one-to-many and many-to-many.

One-to-one is very rare, as generally it means that two tables should be combined into a single table. As an example of this type of relationship, however, perhaps you have a table of People and a table of Authors. All of the information you track for a person you also want to track for an author, but you don't want to keep the author-specific fields within the People table. You could create the two tables separately, and include within the Author table a Person ID field. This would be a one-to-one relationship. An author is related to a single person and a person is related to, at most, one author.

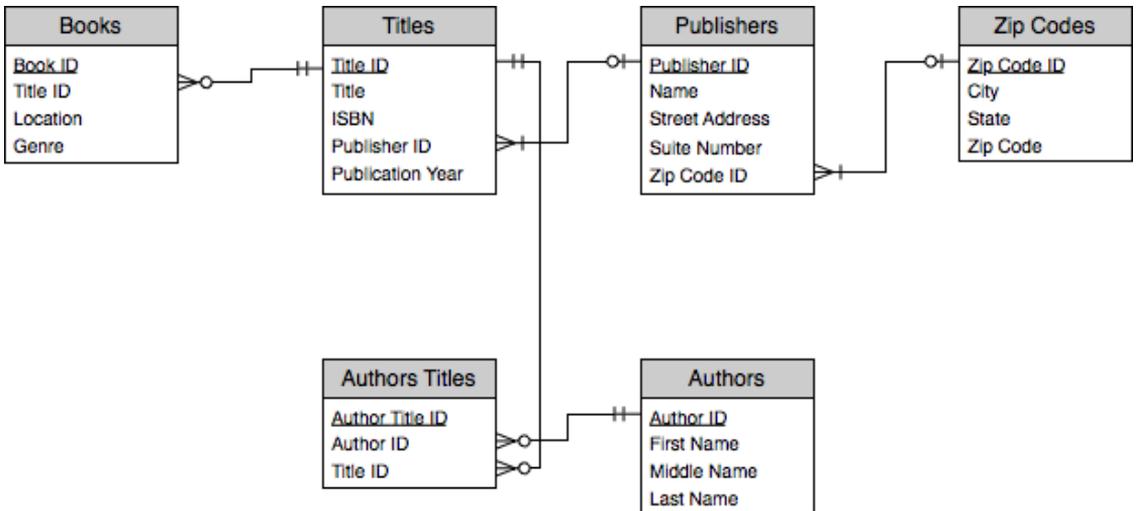
One-to-many is the most common type. Most of the relationships we've discussed fall into this category. A Title can have many Books, but a Book has only a single title. A Zip Code can apply to many Publishers, but a Publisher is assigned a single Zip Code. From a practical standpoint, one-to-many relationships are the only ones you'll probably work with, as when building an actual database, one-to-one relationships are rare and many-to-many relationships are broken into a pair of one-to-many relationships.

The relationship between titles and authors is our example of a many-to-many relationship. An author can write many titles and a title can be written by many authors. However, in our design, we broke this many-to-many relationship into a pair of one-to-many relationships via the Author Title table. The sole purpose of this table is to facilitate this many-to-many relationship via two one-to-many relationships. Any time you have a many-to-many relationship, you'll need to design it in a similar manner.

Entity Relationship Diagrams offer a standard set of arrowheads to help indicate the type of relationship. Here's our ERD with the addition of these arrowheads. The crows feet on one side of each of the lines indicates which of the tables is the "many" in the one-to-many relationship.



Another useful attribute of a relationship to document is which relationships are required and which are optional. For example, for our purposes, it doesn't make much sense to have a Book without a related Title, so each Book must have a related Title. On the other hand, it's possible to have a title in our database without a corresponding book. Perhaps it's on our wish list and we haven't bought it yet. So a Title may have one linked Book, many linked Books, or no linked Books. Our ERD now looks like this after considering the same circumstances for the other relationships.



Here's how to interpret the symbols. A circle with crows feet means that the table linking to the connected table can have zero, one, or many related records. A line with the crows feet means that there must be at least one related record, but there could be many. Two lines mean that there must be a related record, whereas a circle with a line means that there may be a related record, but need not be, although if there is a related record, there is only one.

Concretely, a Title may have a related Book, but need not. A Book must have a related title. Alternatively, a Title may have zero, one, or many related Author Title records (perhaps we don't know the author). Similarly, a Publisher may have a Zip Code, but need not have one (perhaps we don't know the publisher's address).

At this point, we've completed the initial design, and can move forward with implementing the design.

Standards and Conventions

Before we get into how to implement our design in FileMaker, a word about standards and conventions. Standards, in this context, means the naming and organization conventions for FileMaker objects such as fields, relationships, layouts, scripts, etc.

What you're about to see is one individual developer's standard when building a database. The standards I use work for me and have evolved extensively while I've been working with FileMaker. One's choice of standards sometimes become a religious war between developers. The point of naming and organization conventions within FileMaker is consistency within an application. FileMaker systems can get quite complex, with dozens of tables, each with dozens of fields, layouts, and scripts, resulting in hundreds, if not thousands, of named objects.

To give you an example, we've seen that each table should have a primary key field. My own convention for naming this is ID (short of identifier). If a table needs to have a relationship to another table, and therefore a field for that table's primary key (called a foreign key, such as the Title ID field found in the Books table), I simply prepend the other table's name to the field name, as in `TitleID`. An alternative convention is to prefix the field name with `pk_` for the primary key, as in `pk_book_id`, and `fk_` for foreign keys, resulting in `fk_title_id`.

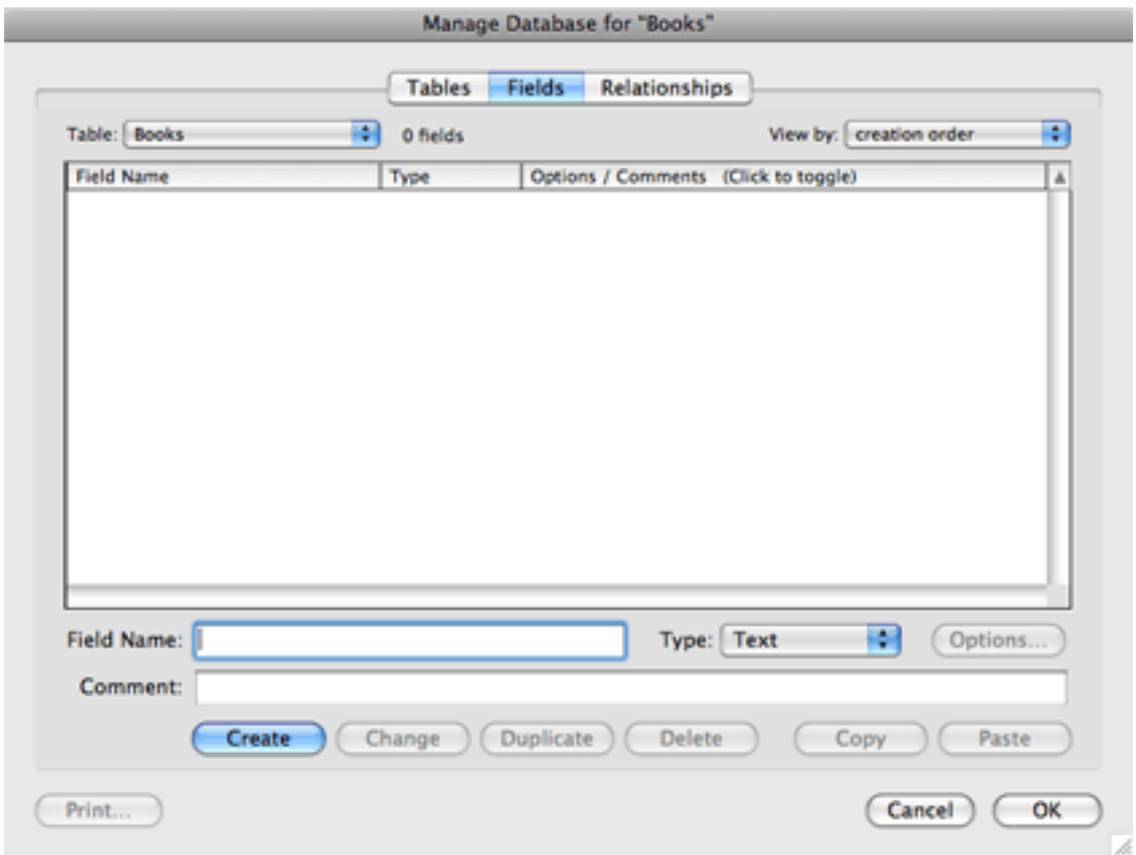
The point is not that one is better than the other. Use whatever works for you in the end. The point is, be consistent, at the very least within a single system, and ideally across all the systems you build. I won't be explicitly pointing out all of the conventions I use, as they are going to be pretty obvious as we move forward.

Implementing the Design

We have our design, so now we're going to build the system we've designed in FileMaker, creating the tables and fields. I'll be using FileMaker Pro 9 Advanced as I perform the necessary actions, but for the time being, everything mentioned should work almost identically in FileMaker 7, 8, or 8.5.

Launch FileMaker. If you've been using FileMaker for a while and have turned off the Quick Start feature, you'll get a standard save dialog box. If you get the Quick Start window, click the "Create empty database" radio button and then click "OK," at which point you will then see the standard save dialog box.

Select where to store the database, and name the new file "Books.fp7". FileMaker will create the empty file, add a single new table to it called **Books** (after the name of the file), and bring you to the Manage Database window where you can create fields for this table.



We're simply going to use the fields we've come up with from our design. The first one is the primary key field. Click in the Field Name text field and type ID. From the "Type" pop-up menu, select Number (or press Command-N). If you like, enter a comment to the effect that this is the primary key field for this table. Finally click "Create" or press Return.

Since this is the primary key field, we need to change its behavior to ensure that each record gets a unique value. Click the "Options" button.

By the way, if you're more comfortable using the keyboard to navigate within the interface, you can get to the options in an alternate route. After creating the field, you'll find yourself back in the "Field Name" text field. Pressing Shift-Tab will take you to the field list, with the newly created field being highlighted. Pressing the space bar at this point will bring you to the "Options" window.

Click the checkbox labeled "Serial number," leaving the default options as "Generate: On creation," "next value" as 1, and "increment by" also as 1.

This means that each time a record in the **Books** table is created, the system will automatically assign the next value listed here to this field, and increment the next value by 1. So the first time we create a record, this ID field will have a value of 1, the second time, a value of 2, and so on. There are other ways to populate a primary key field, but this is the most straightforward and simplest.

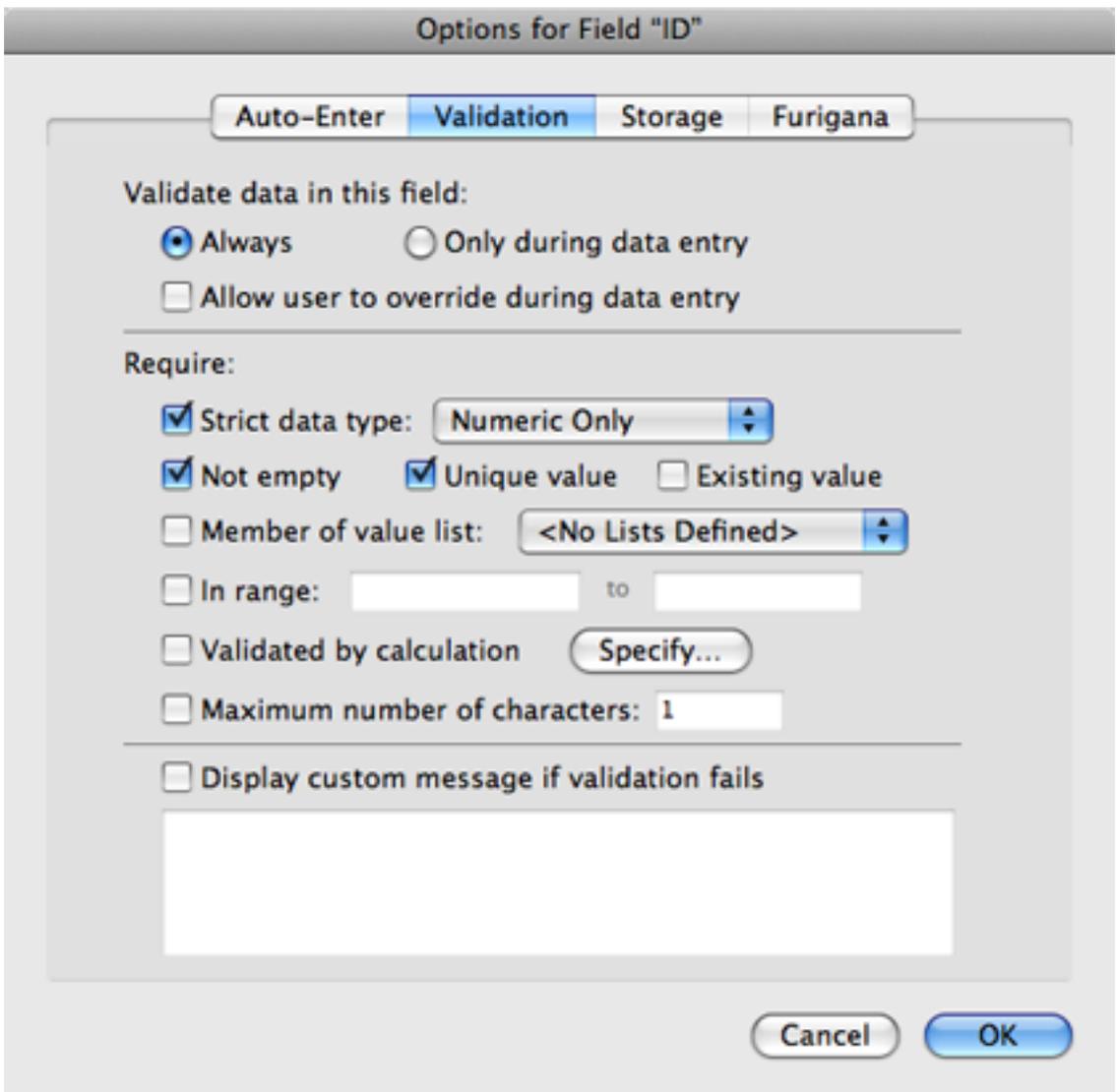
Also remember that a key field should not change its value. Therefore, to protect against this, click the checkbox labeled “Prohibit modification of value during database entry.”

The image shows a dialog box titled "Options for Field 'ID'". It has four tabs: "Auto-Enter" (selected), "Validation", "Storage", and "Furigana". Under the heading "Automatically enter the following data into this field:", there are several options:

- Creation: Date
- Modification: Date
- Serial number
 - Generate: On creation On commit
 - next value: 1
 - increment by: 1
- Value from last visited record
- Data: [Empty text box]
- Calculated value: Specify...
- Do not replace existing value of field (if any)
- Looked-up value: Specify...

At the bottom, there is a checkbox Prohibit modification of value during data entry. At the bottom right are "Cancel" and "OK" buttons.

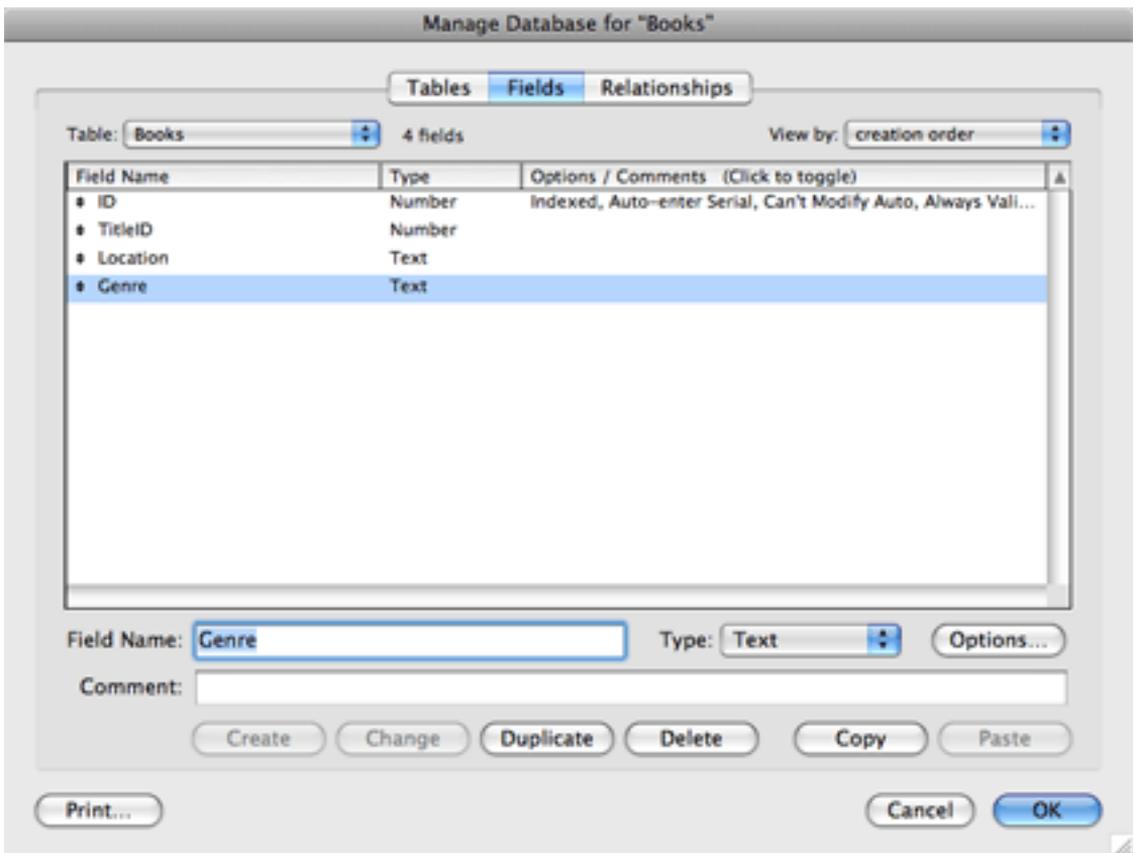
Additional precautions can be taken to ensure that each record has a unique value and that there are no blank values. Click the “Validation” tab in the “Options for Field” window. Click the radio button labeled “Validate data in this field: Always” and uncheck the checkbox labeled “Allow user to override during data entry.” Click the checkbox for “Strict data type;,” leaving the pop-up menu option next to it as “Numeric Only.” Check the boxes for “Not empty” and “Unique Value.” Click “OK” to save your changes to the field options.



None of these should ever be an issue given the options that we specified in the “Auto-Enter” tab, but they do provide an extra level of integrity upon the ID field. Also, if we end up importing records from another source, this additional validation will help us be sure that the imported data doesn’t break the integrity of our existing data.

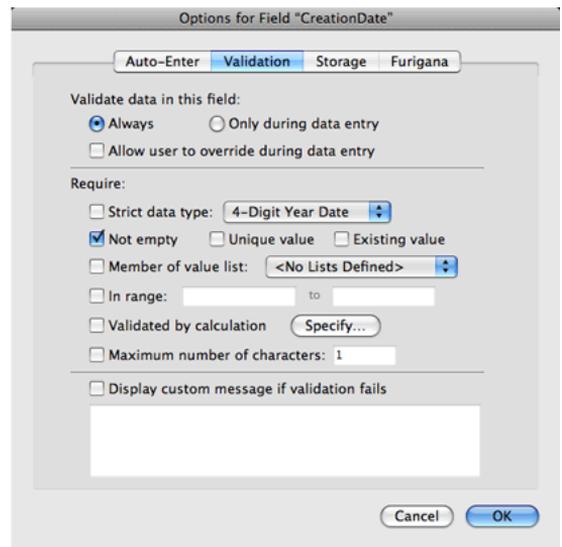
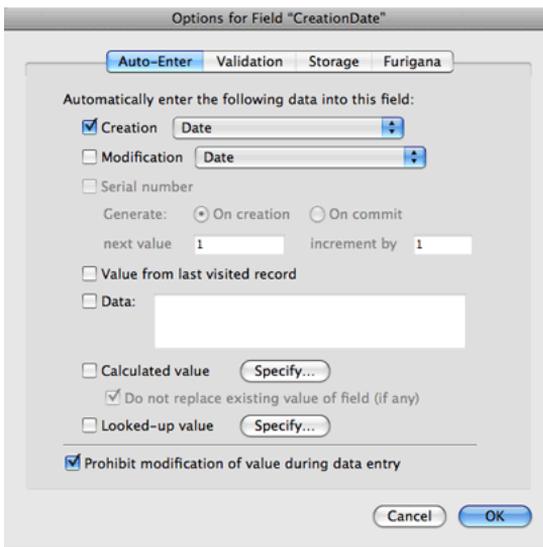
Create another field called `TitleID`, also a number field. This one doesn’t need any further options set.

Now create two text fields, one for `Location` and one for `Genre`. You can set the field type to Text by either selecting “Text” from the “Type” pop-up menu or by pressing Command-T. Neither of these need to have any options set.

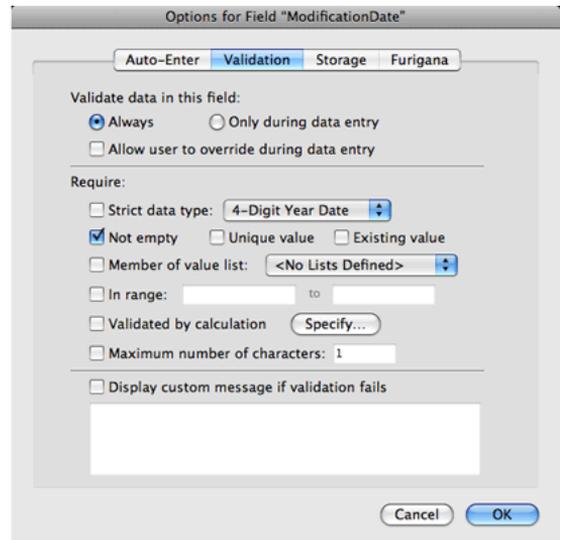
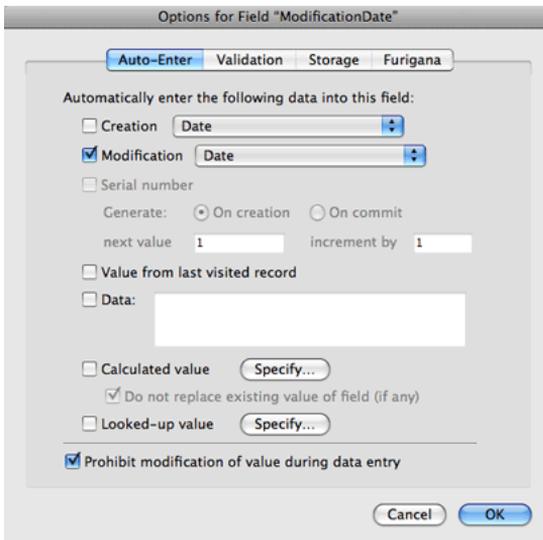


A few more fields will be useful later, and it's best to create them immediately. These are utility fields, allowing simple record tracking when a record was created. We could postpone their addition until we see an actual need for them, but by that point it may be too late, as it may be after many records have already been created, and adding the fields at that point would leave those existing records without the creation or probably the modification information.

The first field is **CreationDate**. Specify that it's a Date field by selecting "Date" from the "Type" menu or pressing Command-D and set its options to auto-fill with the creation date. Prohibit the modification of this field's value and set the validation to always validate, disallowing the user to override the data, and check the box to ensure the field is not empty. Then click "OK" to save the changes to this field.

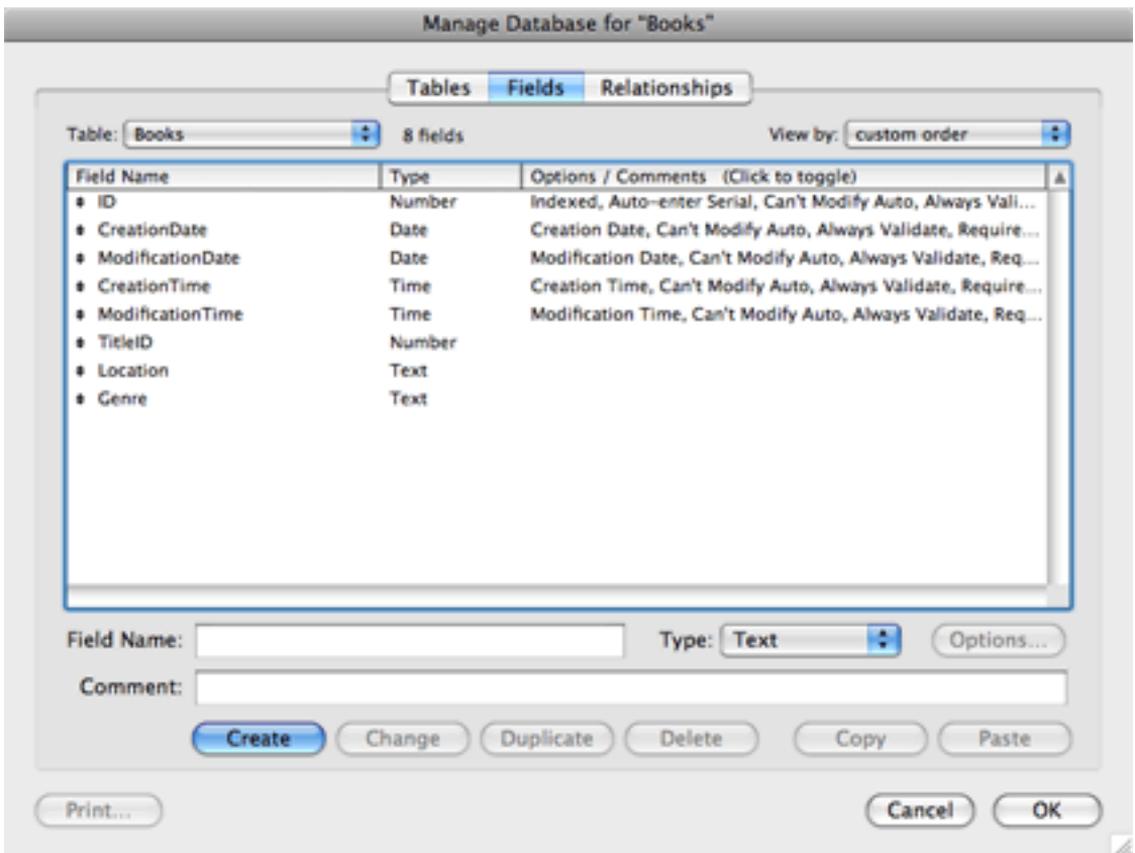


Create the field for `ModificationDate`, with identical options to the `CreationDate` field, but this time setting it to auto- enter the modification date.



Create two additional fields, `CreationTime` and `ModificationTime`, similar to the date fields, but with the “Type” as “Time” (use Command-I to set a field type to Time via the keyboard), and the appropriate auto-enter options.

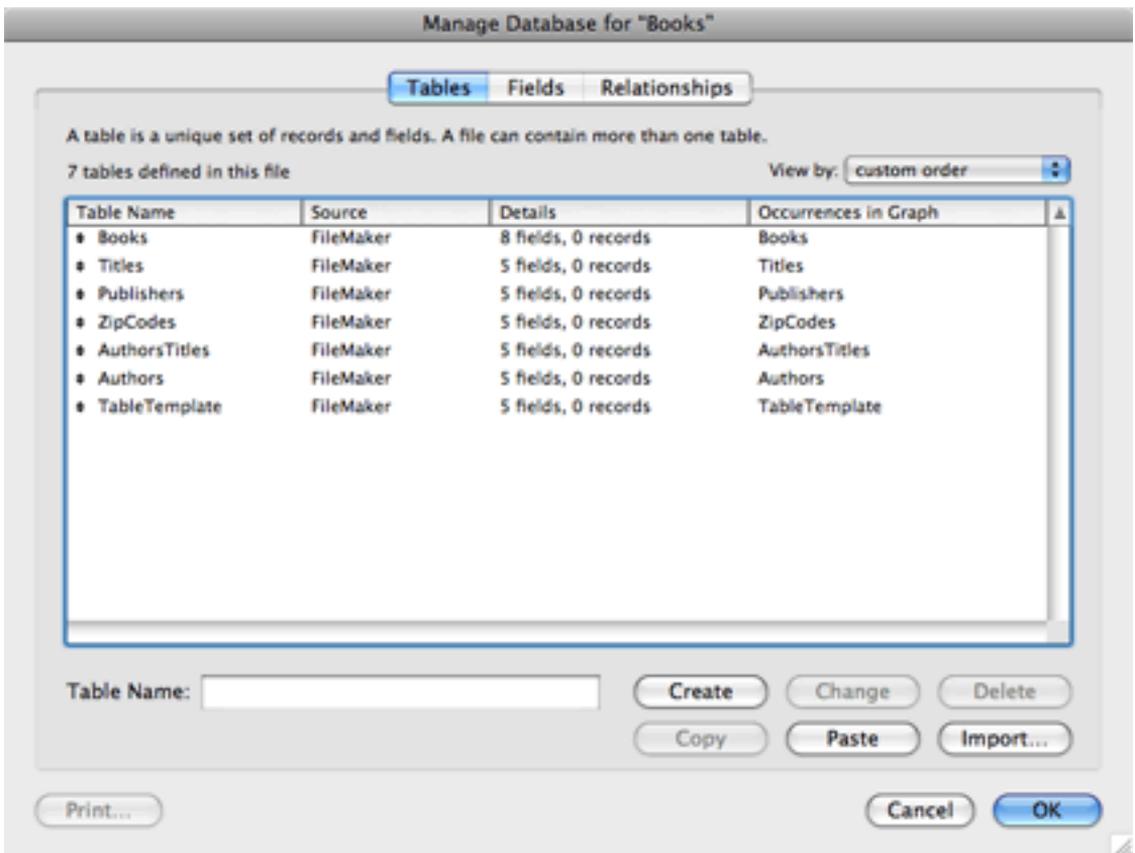
I find it useful to keep the utility fields next to each other. You can reorder the fields in the list by dragging the double-sided arrow to the left of each field name. Once you have done this, your field list should look like this.



We have eight fields in our `Books` table, and five of them should be in all of our tables. So the next thing we're going to do is duplicate the table we have built, remove the Book-specific fields and duplicate this table template for the rest of our tables.

Click the “Tables” tab and make sure the `Books` table is selected. Click the “Copy” button or press Command-C. Either click the “Paste” button or press Command-V. You should see a new table called `Books 2`. Highlight the table name in the field at the bottom of the window and rename it `TableTemplate`, clicking the “Change” button after doing so (or, from the keyboard, press the Tab key and then the up arrow key, pressing Return to accept the default choice to save the change). Double-click the `TableTemplate` table. Click on the `TitleID` field, and then Command-click on `Location` and `Genre`. Click the “Delete” button or press the Delete key on your keyboard to remove these fields, and then click the “Delete” button in the confirmation dialog that appears.

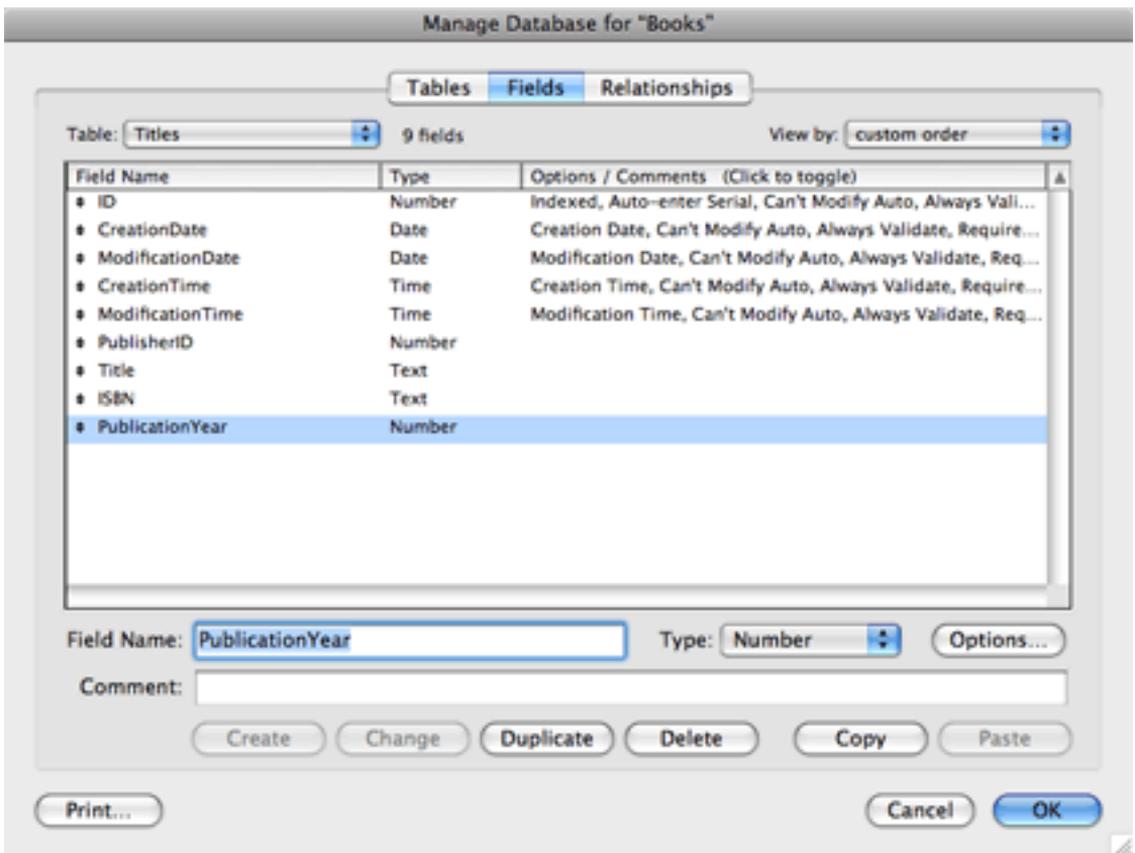
Return to the “Tables” tab and duplicate the `TableTemplate` table by copying once and then pasting five times, renaming the resulting new tables `Titles`, `Publishers`, `ZipCodes`, `AuthorsTitles` and `Authors`. Reorder the tables so that the `TableTemplate` table appears at the bottom, as it may prove useful to have if we add more tables in the future.



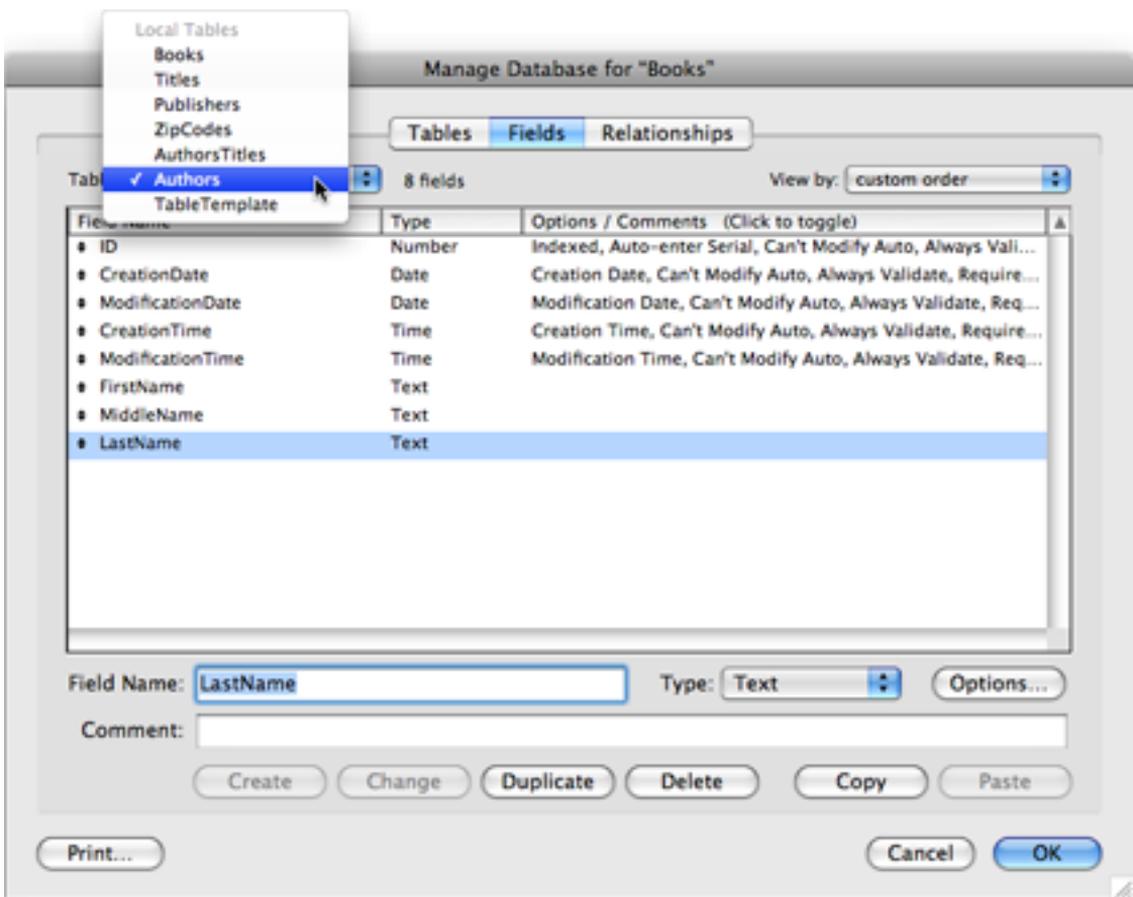
On a side note, this is one reason I always name my primary key field ID. Creating a table with the default fields becomes trivial by doing so.

Copying and pasting tables in this manner is considered an import to the FileMaker application (you may have noticed the "Import" button also found in the "Tables" tab, which allows you to import a table from another file). When FileMaker performs an import of tables, it keeps a log of the progress in case there are problems. Therefore, if you check the folder that stores the "Books.fp7" file, after pasting the tables, you'll find a new file called "Import.log." Go ahead and open it if you're curious, but you can trash it with no ill effects.

Let's create the table-specific fields in the rest of our tables. Double-click on the **Titles** table and create the fields for **PublisherID** (number), **Title** (text), **ISBN** (text), and **PublicationYear** (number).



To go directly to the Authors table, select it from the "Table" pop-up menu at the top left of the window. Create fields for `FirstName`, `MiddleName`, and `LastName` (all text fields).



Move on to the `AuthorsTitles` table, creating fields for `AuthorID` and `TitleID`, both number fields. In the `Publishers` table, create the fields for `ZipCodeID` (number), `Name` (text), `StreetAddress` (text), and `SuiteNumber` (text). Finally, move on to the `ZipCodes` table, creating the fields for `ZipCode` (text), `City` (text), and `State` (text).

Some Explanation

For most of the fields we've created, the field type is obvious. An author's first name is obviously storing text. But what about the zip code? These are numbers, so perhaps the `ZipCode` field in the `ZipCodes` table should be a number field.

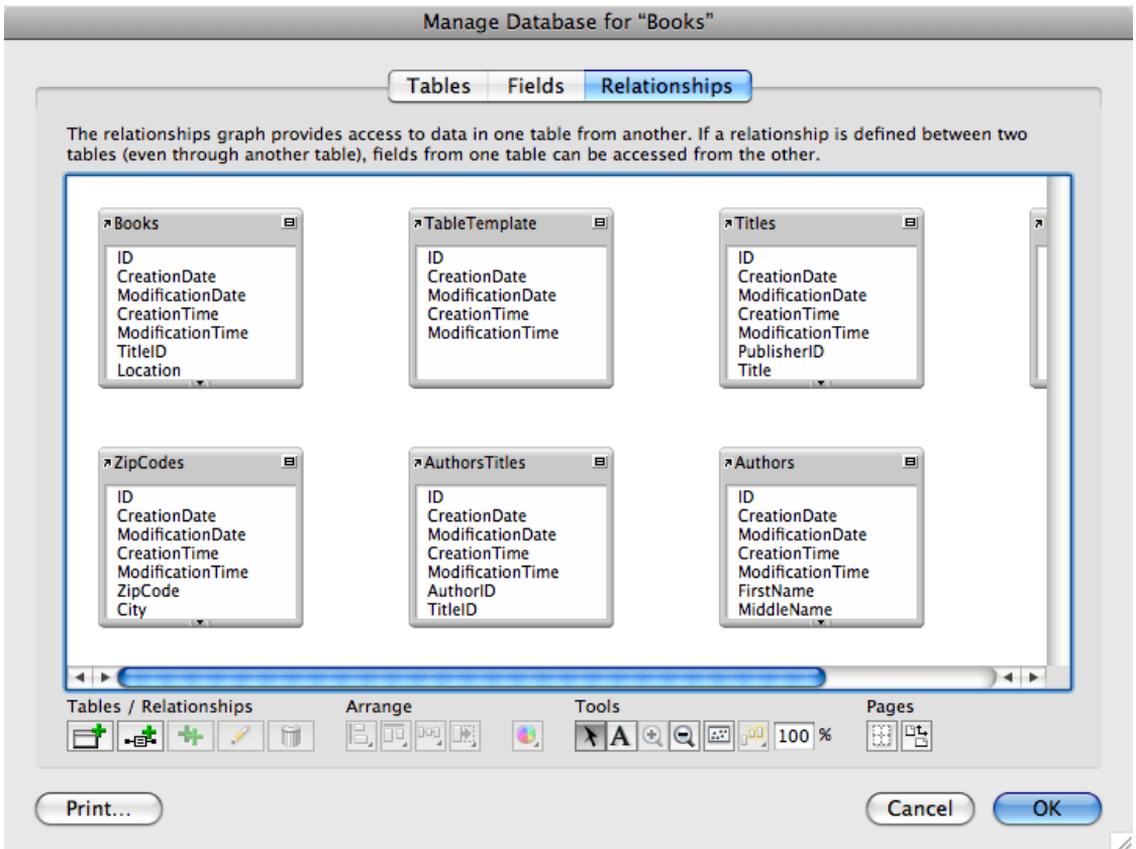
While zip codes are represented by digits, they aren't numerical data. One would never perform mathematical operations on a zip code, and some zip codes begin with zeros, which would be removed in the display of zip code data if we stored it as a number field. The basic criteria for storing data as a number is whether you anticipate the possibility of performing mathematical operations on the data. Zip codes fail this test, while the year a title was published satisfies it. One could imagine wanting to calculate how long it's been since a title was published.

What about our primary key fields? One wouldn't normally perform mathematical operations on these key fields, so why store them as number fields? Simply put, it doesn't matter

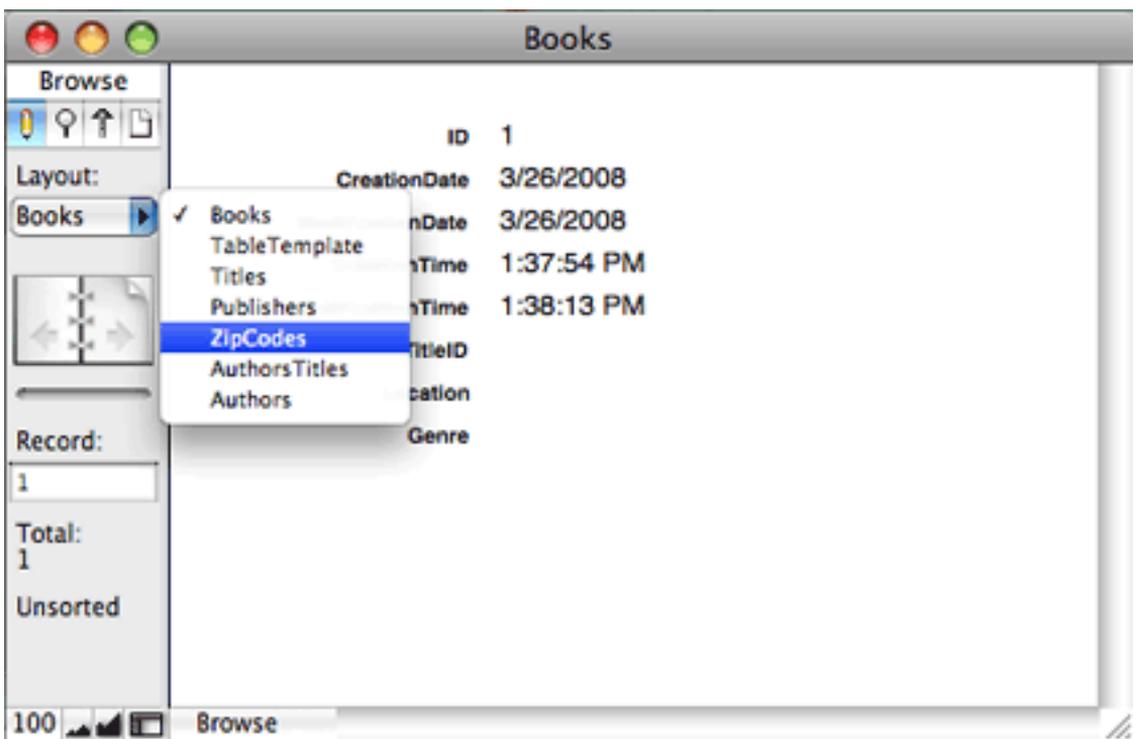
for storage, but for sorting. If I ever want to sort these fields by the order in which they were created given their primary key value, storing their values in number fields will give the proper results. Storing them as text would do things like sort the 100th record before the second.

Table Occurrences

While creating these tables, you may have noticed the column in the Table tab labeled “Occurrences in Graph.” Each new table created what’s called a table occurrence. We’ve not yet created any relationships, so the utility of these is thus far limited. You can take a look at them by clicking the “Relationships” tab.



For now, we’re done defining our database, so click the “OK” button to save all of these definitions. After doing so, you’ll find that in addition to the automatic table occurrence creation, FileMaker has also created automatic layouts for each of our tables, each with all of the fields we defined for each table. It also created the first record in the Books table because that was the first table created.



At this point, we have a perfectly usable database. We can create records for any of our tables and move between tables and records using FileMaker's built-in interface elements (such as the layout menu shown above or the book icons to move between records). However, linking data between tables must be done manually, which would prove difficult and tedious as the database grows.

Next time, we'll refine our database, using the layout tools to make it look better and using other features of FileMaker to make data entry easy, especially when linking records in one table to those in another.

Copyright © 2008 Charles Ross, cross@atpm.com. Charles Ross is a Certified FileMaker 7 Developer and the Chief Technology Officer of [Chivalry Software, LLC](#), a company specializing in custom database, web and automation software and publisher of [Function Helper](#), a FileMaker calculation debugging tool. He was a contributing writer and the technical editor for [The Book of FileMaker 6](#) and has contributed to [ISO FileMaker Magazine](#) and [Macworld](#) in addition to his [series on AppleScript for ATPM](#).



Desktop Pictures

by ATPM Readers

Pictures From ATPM Readers

We need new desktop pictures each month. [Write](#) to us!

This Month's Desktop Pictures

This month, we feature a collection of desktop pictures from ATPM readers.



[View Pictures](#)

Isweeney submitted a photo of a sunset over the Caribbean Sea, about one day west of Jamaica.

Hector Ramos submitted three photos from Puerto Rico.

Morgan Bowe submitted some photos from a trip to Singapore and Bintan Island, Indonesia, and a photo from Rosebud, Victoria.

Narcis Parfenti submitted two photos of Tulcea, Romania at dusk.

Ken Zindle submitted a photo of some white birds on a beach.

Previous Months' Desktop Pictures

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

Downloading All the Pictures at Once

Some browsers can download an entire set of desktop pictures at once.

iCab Use the Download command to download “Get all files in same path.”

OmniWeb Choose “Save Linked > Images...” from the File menu.

Safari Use [this Automator workflow](#).

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 through 10.5.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









Copyright © 2008 Matt Johnson, mjohnson@atpm.com.



Book Review

by Tom Bridge, tom_bridge@mac.com

The Book of Wireless, 2nd Edition

Publisher: [No Starch Press](#)

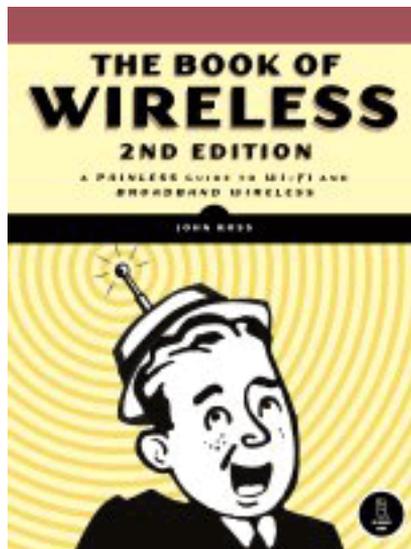
Author: John Ross

Price: \$30

Trial: [Chapter 2](#)



When Arthur C. Clarke spoke about sufficiently advanced technology appearing to the layman as magic, there are many days that I'm certain he's talking about about wireless networking. Click a button on a menu bar, type in a password, and wham, there you are, on the Internet. A button. Not a cable, not a router, no modem, no dial-tone, just a software button and there you are. Have you ever wondered, though, what's happening when you click that button?



I've spent the better part of the last decade accepting the miracle of wireless networking and connectivity. I understand the basic concepts at play, but it's nice to see a book assume very little and provide an immense amount of detail, so the reader gets a better picture of how networking works. Many books that provide information about networks and the Internet are laced with jargon and impenetrable language, and in this regard, Ross delivers us from the twin evils of obscurity and obfuscation. The terms are clear, the text well supported with good graphics, and the concepts well-explained.

Don't expect to skip around, though, and be sure to dive in at the beginning and continue to the middle, instead of vice versa. The concepts in the book are cumulative, and if you

pick up and read chapter 3 on “How Wi-Fi Works” without getting the base concepts in order, you might wind up feeling a bit confused. Unlike the other No Starch book that every sysadmin should own, [Cisco Routers for the Desperate](#), *The Book of Wireless* isn't about emergency knowledge or handy reference. Instead, it's about teaching you more than just mastering the magic, and about learning how it's performed.

But how does this relate to Macs, you ask? Aside from being compelling good knowledge for the general computer user, there's an excellent section dealing with Apple's AirPort technology, both hardware and software, that makes an excellent companion for anyone using an AirPort Base Station. It covers in detail the various applications of wireless networking for the Mac: setting up your home network, setting up good security, and basic operation. It does fall short in failing to address the additional features that the AirPort Express and Extreme make use of, including AirPort Disk and Printer Sharing, which I think could have used more explanation.

In addition, the section on Virtual Private Networking (VPN) does not cover either the Tiger or Leopard methodologies for connecting to a VPN network as part of proper security on a public wireless connection. Given the emphasis in the rest of the book on proper security at the machine level, it's disappointing that even Unix gets a better shake than Mac OS X for VPN use. The depth of understanding of the whys and hows of wireless networking, though, are certainly nothing to scoff at, and why this book has a permanent place on my shelf, and why it belongs on yours.

Copyright © 2008 Tom Bridge, tom_bridge@mac.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Newer Technology iPhone Accessories Roundup

Bass Response Earbuds

Developer: [Newer Technology](#)

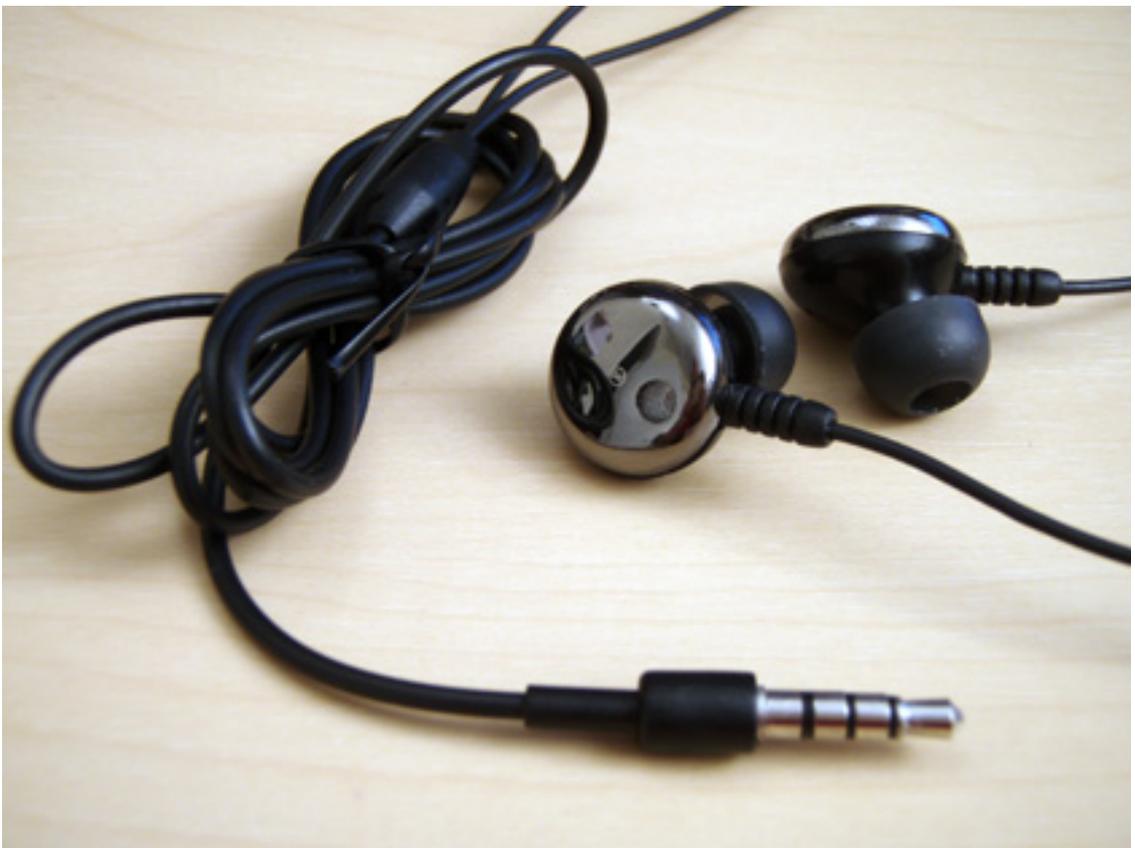
Price: \$20

Requirements: iPhone

Trial: None



In my experience, most any pair of ear canal headphones produces a somewhat muffled sound. Perhaps it is the shape of my ears that affects the sound, since these particular headphones are still reasonably popular.



Before I continue, allow me to lock down some terminology that relates the physical differences between two different styles of earbuds. Though Newer Technology (and other

companies) refers to these headphones as “earbuds,” they’re really best referred to as “ear canal headphones.” Conversely, the term “earbuds” should refer to the type of headphones that rest just outside the ear canal, such as the kind Apple ships with iPods.

Sound quality aside, I generally find ear canal headphones to be more comfortable than traditional earbuds, and Newer Technology’s Bass Response headphones are no exception. Moreover, these ear canal headphones sound better to me than many others I’ve sampled. As the Bass Response name indicates, they have pretty good bass, while not muffling the higher tones to a horrible degree.

There’s a very specific reason I rate these headphones as Okay instead of Good or Very Nice. The Bass Response Earbuds that I received had three rings on the audio plug tip. The third ring is only supposed to be present if a microphone is in-line, but these headphones do not have a microphone. When an iPhone detects this third microphone ring, it automatically deactivates the internal microphone. Consequently, when using the Bass Response Earbuds, the plug *must* be removed when making or receiving a call, or the other person will not be able to hear.

If Newer Technology updates future production of the Bass Response Earbuds with a two-ring tip, then I might be inclined to recommend the product with a Very Nice rating. However, one other fact bothers me. These earbuds with no microphone are the same price as the version with a microphone (see below). So there’s absolutely no incentive to choose the non-microphone model.

Hands-free Mic and Earbuds

Developer: [Newer Technology](#)

Price: \$20

Requirements: iPhone

Trial: None



The headphones portion of this product is identical to the Bass Response Earbuds (see above). Thus, my comments regarding sound quality would be repeated for the Hands-free Mic and Earbuds product.



Unlike the Bass Response Earbuds, though, this item does have an in-line microphone for use without unplugging the cord. Strangely, it sells for the exact same price as its microphone-deprived cousin. For this reason, there is absolutely no rationale for choosing the Bass Response Earbuds.

Those who are keen-eyed may notice that the photo on Newer Technology's Web page for the Hands-free Mic and Earbuds shows only two rings on the tip and no apparent in-line microphone. My product photo (above) clearly shows that there is, in fact, a microphone as well as three rings on the tip. My guess is that Newer Technology's product photo is mistakenly showing the no-microphone ear canal headphones. Furthermore, since this photo shows only two rings on the tip, I suspect it may simply be a manufacturing error that several, including the set I received, were mistakenly shipped with a three-ring tip.

The improper product photo notwithstanding, I was quite pleased with these headphones and have all but permanently put away my Apple earbuds.

Mic Extender Cable

Developer: [Newer Technology](#)

Price: \$15

Requirements: iPhone

Trial: None



Any headphones or earbuds—with or without a microphone—can be used with Newer Technology's Mic Extender Cable. An iPhone will utilize the cable's microphone for calls regardless of whether a two- or three-ring tip is inserted into the female end.



I found nothing to warrant anything less than a top rating for the Mic Extender Cable. Various people I called reported that this cable's microphone sounded better than the iPhone's built-in mic. The toggle button to answer or disconnect calls is easy to find without looking, and the plastic clip for attaching the mic to a shirt hem is very well designed.

Headphone Jack Adapter

Developer: [Newer Technology](#)

Price: \$8

Requirements: iPhone

Trial: None



I really wanted to give top rating to Newer Technology's Headphone Jack Adapter. After all, it does the intended job very well. It's also priced \$2 less than Griffin Technology's adapter and \$3 less than Belkin's.



Unfortunately, the aforementioned three-ring tip problem is also present in this adapter. If not for this issue, I would use my Bose QC2 headphones and this adapter to enjoy music on my iPhone. There's no reason I shouldn't be able to listen to a caller through my Bose cans while simply holding the iPhone up to my mouth so the caller can hear me. The Mic Extender Cable (see above) with its in-line microphone solves the problem.

In the likely few cases when someone owns a preferred set of headphones that *does* have an in-line microphone, Newer Technology's adapter is the best choice, since it would carry the microphone's signal through the third ring. Those individuals might feel this product deserves an Excellent rating.

I chose Very Nice because my impression is that most customers shopping for this type of adapter are using headphones without microphones. Thus, a two-ring adapter would be desired, such as any one of the aforementioned competitors' products.

Auto Charger

Developer: [Newer Technology](#)

Price: \$13

Requirements: iPhone

Trial: None



Newer Technology utilized a shaft design for the Auto Charger that is shorter than most vehicle power port devices. I like this design since it doesn't protrude as far from the power port. However, three design aspects would cause me to shop for a different charger.



Most significantly, a considerable amount of force is required to insert and remove the charger's plug—a lot more than other chargers I've used. This overly tight fit is not required for the plug to stay in place. It's a good thing the end of the plug is flared so as to provide something to grip firmly while yanking it out of a power port. It's very much needed.

I prefer some sort of visual indicator on any car charger device to show that power is flowing. Newer Technology's Auto Charger has none. Such an indicator is helpful in determining whether a particular car still provides power while the key is out of the ignition.

Some people may like the all-in-one design, but for iPhones and iPods, a car charger that uses a standard USB cable attached to the charger plug makes a lot more sense. On more than one occasion, I have needed to charge and sync my iPhone at a remote location when I had my laptop with me, but no other cables. I simply absconded with the USB-style cable from another brand of car charger to do the job.

As far as vehicle power port chargers go, Newer Technology's Auto Charger will do the job. Potential customers should be aware of these design issues, which can have an impact on its convenience and practicality.

Copyright © 2008 Lee Bennett, lbennett@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Software Review

by Ed Eubanks, Jr.

PhotoAcute Studio 2.77

Developer: [PhotoAcute](#)

Price: \$19 (Portable); \$49 (Standard); \$119 (Professional)

Requirements: Mac OS X 10.4.8, 512 MB RAM (1 GB for RAW images).

Trial: Feature-limited



I've enjoyed photography ever since my father brought home a camera that none of the employees in his office could figure out; he and I eventually found photography to be one of many hobbies that we enjoyed together, and I've even done it professionally over the years.

When I made the jump to digital a few years ago, one of the limitations that I felt almost immediately (and that had kept me from jumping in earlier) was the latitude that digital photography offers in exposure: a little too much or too little light and your image will suffer considerably. While very much like shooting transparency/slide film, it's nothing like the print film I was used to.

When I saw PhotoAcute Studio, I hoped that I had found a solution to this with the less-than-perfectly exposed images I have shot over the years. Here's what I found...

Overview

PhotoAcute Studio sets out to do much of what I hoped for, and more: increase image resolution, noise reduction, geometry correction, expansion of dynamic range, correcting chromatic aberrations, and even removing unwanted objects (like when someone walks through your shot). All of this, without noticeable loss of detail or sharpness.

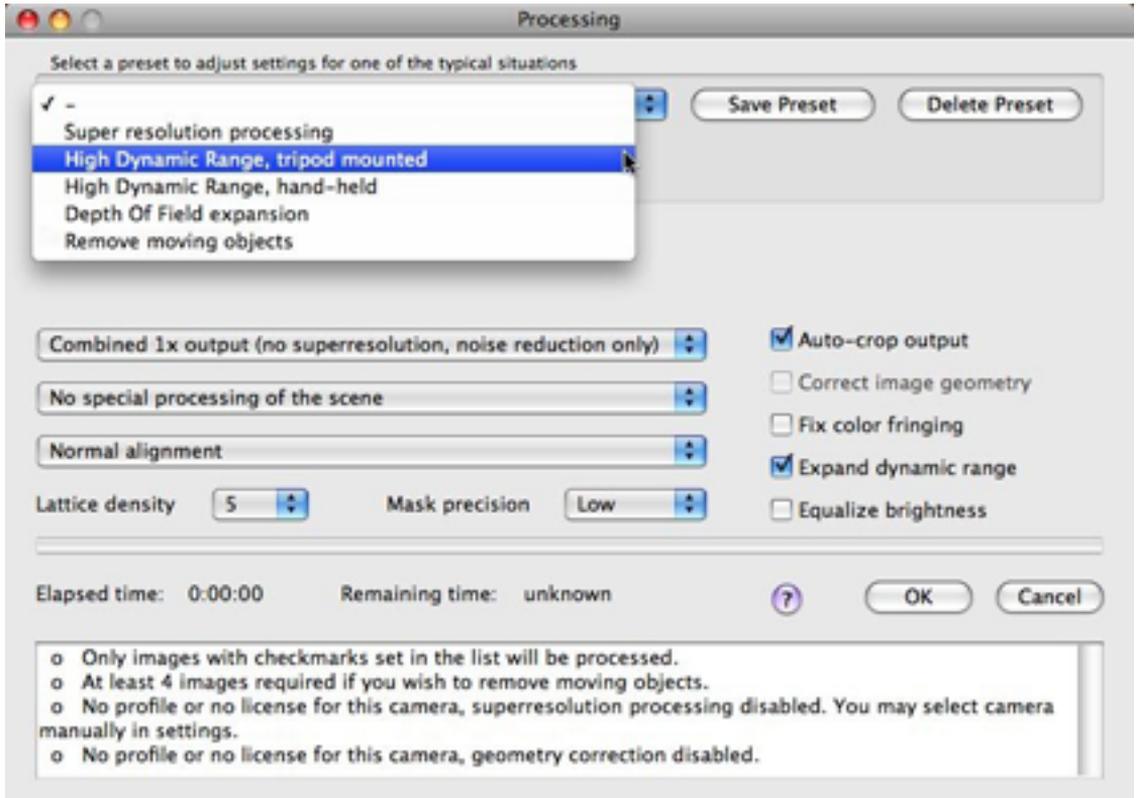
Does this sound too good to be true? Well, it is and it isn't. In a best-case scenario, the application can accomplish all of these capably. The tricky part is finding such a best-case scenario: there are a number of requirements that must be met to pull it off, and they aren't always easy or even possible to accommodate.

For starters, you must have at least two images of the same basic composure. These can be off a bit—a shifted perspective, for example, or one that is slightly blurry or out of focus. But if you don't have multiples, you can't even get started. PhotoAcute Studio assumes that you have been using a burst-mode or some form of rapid-fire shooting, allowing the multiples to compensate for the missing or errant information in each other.

If you have multiple exposures that are very similar, however, you may be on the way to getting the perfect exposure—eventually.

Very Technical

The application itself appears very simple up until you actually begin the processing; at that point, however, a dialog box appears with an array of options that is quite intimidating. There is a lot of power here, and the tools contained in this application offer automation of techniques and processes that would be quite time consuming in Photoshop or a similar application. Make no mistake, however: this is a tool for people who understand photo editing beyond the iPhoto level.



The developers have provided a good summary in the manual/user guide that guides you through what each individual feature offers—but frankly, if you aren't adept at photo editing or familiar with some technical terminology, you may get lost in the manual. Further, the manual doesn't specify how to accomplish certain results; your best bet for finding these is to look at the examples provided on the Web site and try to emulate them.

Works With Phones, Too

A great feature of PhotoAcute Studio is the ability to work with mobile phone cameras. In fact, many of the features seem to be especially suited to adjusting images taken with a mobile phone. (A [separate version](#) is available that installs directly onto your mobile phone, giving access to many of these tools immediately upon shooting. I hope they're tinkering with the iPhone SDK to get a version ready for iPhone users!)

This appears to be a market they are just dipping into: the list of supported devices (as in, pre-defined profiles) is quite limited. I was surprised to find that there were no profiles

for popular phones like the Motorola RAZR, Palm Treos, RIM BlackBerrys, and Apple iPhones. I assume that profiles for these are in the works.

This doesn't mean you can't use PhotoAcute Studio with your mobile phone pics. Just like with regular digital cameras, unsupported devices can make use of the features and abilities, improving photos well.

Frequent mobile phone camera shooters will already realize many of the limitations here: trade-offs in detail or sharpness may be corrected to a helpful degree, though you'll never get them to pass for digital pictures from a 3–6 megapixel model. Still, PhotoAcute Studio can take an almost-decent photo from a mobile phone and help it become an acceptable one.







As you can see in my examples, the first image was improved upon by PhotoAcute Studio (image two); the whole photo is sharper and less noisy, with fewer distracting elements in the background. Still, there are aspects that suffer (notice the loss of crispness on my glove,

for example), and it doesn't improve enough to be better than the third image, which was the re-shot version taken a moment later.

Some Features Not Available

Many will be delighted to learn that PhotoAcute Studio offers similar editing tools for video—in fact, this tool will be a valuable addition to a videographer's toolbox.

Only if you're not a Mac user, however, as PhotoAcute Studio includes video capability for the Windows version (and, it seems, the Linux version) but not the Mac OS X version.

Even with still photos, you may find that some of the features are unavailable to you. If you don't have a profile for your camera, features such as superresolution and geometry correction are disabled. What is more, even if there is a profile for your camera, these features are sometimes unavailable to you: my Nikon DSLR is supported with a profile, but photos shot with it in JPEG format and imported first through iPhoto were not recognized as supported.

License Complexity

PhotoAcute offers three distinct licenses, as mentioned in the summary above. The Portable license is for mobile phone users. The Standard license supports compact and “prosumer” cameras with non-interchangeable lenses and no RAW format processing. The Professional license brings support of cameras with interchangeable lenses and RAW format processing.

I think it is great that PhotoAcute offers such a diverse range of licenses, because obviously not everyone is going to need what the \$119 Professional license offers. Kudos to them, also, for supporting camera phones. They wisely recognized both a need and a market for their product across a spectrum, and I appreciate the licensing structures as they are.

That said, a handful of details about this bug me, to say the least. For one thing, many photographers in the middle category—especially the so-called prosumer (when I started in photography these were called serious amateurs) who is serious about photography but doesn't make money from it—will be frustrated with the lack of support for RAW files. This format was once reserved for the high-end digital SLR, but with improvements in processor speeds and memory costs, many fixed-lens cameras now offer RAW as a compression (or, technically, a lack of compression) option. More software supports it, too—including iPhoto. The bottom line here is, RAW is not just a professional-level format anymore, and I wonder how many true amateurs and hobbyists will feel burned about this. (For \$70, Standard license holders may upgrade to a Professional license.)

Another niggling detail is the hard divide between Portable licenses and other licensing options. The default assumption seems to be that, since I shoot a DSLR, I don't also use the camera on my iPhone. I can understand asking Standard license owners to purchase a Portable license for an additional \$19, but it seems like the Professional license should open up every profile and possibility available. Yet, none of the Portable profiles are available to me in my Professional license.

In general, most users will be happy enough with the Standard version—RAW photos can be exported from iPhoto, Aperture, Photoshop Lightroom, etc. as JPEGs and run through PhotoAcute's process without much trouble. But navigating licenses could be a bit easier.

When It Works, It Works?

I'd like to report that, in the right conditions, I was able to get the application to deliver as promised. The comparison photos on the Web site are compelling; it's certainly an appealing prospect to know that I could shoot with this tool in mind, and eventually get the results I wanted.

The trouble is, I wasn't able to find a set of images (other than the ones displayed above) in all of my library that allowed me to accomplish this. I guess over the years I learned to compensate for the inherent limitations after all—and that's not hurt by the high quality metering and focus mechanisms included in most cameras today. At the time of writing, I didn't have a chance to shoot some that would fit the criteria, and so I can't honestly say that I can vouch for the end result as promised.

Wrap-up

My verdict? Another tool in the toolbox always helps; knowing I have this tool available to me may lend some freedom to my shooting in future outings, and I certainly will try PhotoAcute Studio before writing an image off as a loss.

Still, the difficulty of technical use, combined with the very specific parameters in which this application is designed to work, I can't highly recommend it to everyone. The mobile version is a reasonable price, and would be a worthy buy for those who shoot a lot of shots with their phones—if your tendency is to rely on your phone camera at weddings, birthday parties, and graduations (as seems to be the growing trend), you may find PhotoAcute Studio invaluable. Users of other cameras may find that the offerings here are too specialized to justify the cost.

Copyright © 2008 Ed Eubanks, Jr. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Book Review

by Paul Fatula, pfatula@atpm.com

Take Control of Permissions in Leopard

Publisher: [TidBITS Publishing](#)

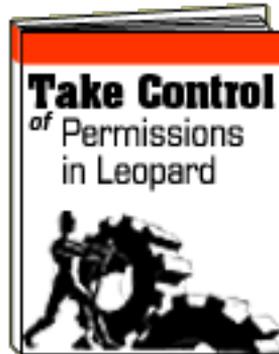
Author: Brian Tanaka

Price: \$10

Trial: [24-page excerpt](#)



I've been using Mac OS X since its first release, and I've been comfortable working with traditional Unix permissions from the command line since the early 90s. But somehow, permissions in Leopard were confusing me. Most of my frustration stemmed from the new Get Info window, which no longer allows the changing of the user and group that owns the file and which, even more strangely, allows multiple users and groups to own a file. It was pretty clear that something new—at least to me—was going on, and I needed to learn about it. I needed to (are you ready?) take control of permissions in Leopard.



It's Not a Book, It's an E-book

Take Control of Permissions is provided as a downloaded PDF. It's not a password-protected PDF, it's not a DRM-wrapped PDF, it's just an ordinary PDF: I can open it easily, anywhere, without restrictions. Yay TidBITS!

True to the electronic age, the e-book is also designed for on-screen reading. That gave me pause, because I always print manuals. Seriously, I've got hundreds of pages of at least half a dozen different Leopard Server manuals printed out and neatly comb-bound sitting on my desk for reading and reference, because that's just what I'm most comfortable with. But OK, I decided, I'll play along: and so *Take Control of Permissions* is the first book I've ever read entirely on-screen.

Gazing at my monitor, my attention wandered a bit more than it does while reading a printed book. Though I was less comfortable in front of my screen than I would have been in a lounge chair, being right at the computer encouraged me to try things as I read, which was a real plus. The ability to search for a term instead of having to hope it's in the index is also great. . .but there is no index, which seriously limits the usefulness of this e-book if it is printed out. When flipping back to an earlier section, nothing in Preview can come close to the convenience of sticking a finger in a book to mark my place.

What's Inside

Having previously been under the impression that everything I needed to know about permissions was contained in a few Unix commands and the Finder's Get Info window, I was surprised that Brian Tanaka had been able to fill 87 pages with information about permissions. From the meanings of owner/group/other to instructions for calculating permissions in octal, the e-book is packed with useful information. As early as page 3, I learned something new. You can manage groups from System Preferences now? Cool!

The e-book takes its reader to the Terminal surprisingly early. I like that a lot, and I think it makes sense to start at the Terminal considering the subject matter. Granted, some users have an aversion to typing commands instead of pointing and clicking, but there's nothing to be afraid of.

Though it turned out that Leopard's Get Info window really is that bad (I'd been holding out hope I was missing something obvious) and third-party applications or the command line must be used for a lot of working with ownership and permissions, I learned a lot about the big change in Leopard: Access Control Lists are now enabled by default. It's a whole new layer of functionality that adds on to the standard Unix permissions I was used to. *Take Control of Permissions* did a great job of laying out how they work and the advantages to using them.

Default permissions is another topic that often causes confusion among users. Though I'm still left thinking "There's got to be a better way!," *Take Control of Permissions* does a good job describing how they do work, with easy-to read tables for reference purposes. There's also an interesting discussion of how the Shared folder works, and why it might not be as good for certain purposes as its name makes it sound.

By the time I got to the "Learn Advanced Unix Techniques" section toward the end of the e-book, I'd already seen most of the relevant Unix commands mentioned and even used on preceding pages, and the "Why use Unix?" question had long since been answered. But the section gives additional information about several commands and gives additional examples of their use. The entry for `chflags` was disappointing, however, saying only that the command is "unspeakably obscure." Tamaka is half right: it's definitely obscure, but you can still talk about it, and what better place than in the Advanced Unix Techniques section of an e-book about permissions?

Conclusion

It's fair to say that most Mac users can learn something useful from *Take Control of Permissions in Leopard*. As nice as it would be if permissions could just take care of themselves and work right on their own all the time, that's not the world we live in. If your Mac has multiple users, sooner or later you'll need to know how permissions work, and this e-book does a fine job explaining them.

Copyright © 2008 Paul Fatula, pfatula@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Book Review

by Ellyn Ritterskamp, eritterskamp@atpm.com

Wikipedia: The Missing Manual

Publisher: [O'Reilly Media](#)

Author: John Broughton

Price: \$30

Trial: [Table of Contents](#)



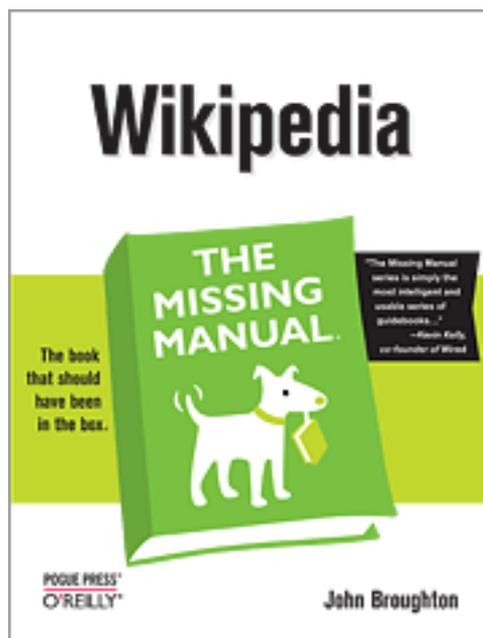
The Missing Manual series is nearly universally praised. The books provide instructions and guidance on software and other computer-related products that don't come with manuals. This book continues the campaign.

Wikipedia is a contemporary sign of our times, a conglomerate, a composite, a melding and mishmash of people and ideas. I wrote about it [two years ago](#), in an attempt to figure out whether this community reference thing was going to work.

It works.

Wikipedia is not perfect, by any stretch. But it is useful and energizing and handy. The notion that you or I can contribute to someone else's understanding of a topic is exciting and rewarding. But don't count on it for verifying anything important. The openness of the system means it's still possible for vandals to monkey with information. It normally gets corrected fairly quickly, but you might be there during the time it's wrong, so be careful. I corrected a page a few months ago that said one of my favorite 85-year-old actors had purchased a condom instead of a condominium.

All that said, if you want to start editing articles and writing them yourself, John Broughton's book is certain to make your experience a better one. I don't say you must own it; much of the information in it you can figure out on your own, but the book would save you trial and error. Much of the information about the site itself is found in the toolbar on the left side of all Wikipedia pages, following a link called [About Wikipedia](#). If you intend only to edit a few pages here and there, likely you will find enough guidance there, and not need the book.



But for folks who enjoy digging into a thing, the book is well worth the investment. I found it new at a large online book retailer for two-thirds of the suggested retail price, but even at full price it's a good buy. Broughton explains the advantages of creating an account, tells you how to read a page's history to see previous edits, and explains the sandbox (a page where you can practice editing without disturbing content). Once you register, you get your own sandbox to play in.

A big chunk of the book is about bumping into and working with other editors. If you plan to invest much energy in this project, I recommend Part II: Collaborating with Other Editors.

Broughton's prose does not include hotkeys, because users approach Wikipedia from many different browsers and operating systems, but there's an appendix listing them. He does provide shortcuts from within the Wikipedia pages, which you can type in the search box on the left side of the pages.

Recommendations

Many casual users will not need the book. If you visit the site more than a few times a day, or find yourself thinking, "I could improve this article because I know something about the topic," or "I wish this was more than a stub—I wish I knew how to expand this into an article," then this book is for you.

Suggestion to Readers

Start with the appendices, especially B and C. They provide useful overviews of how to appreciate the site as a reader, before you start editing.

Copyright © 2008 Ellyn Ritterskamp, eritterskamp@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



FAQ: Frequently Asked Questions

What Is *ATPM*?

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

Are You Looking for New Staff Members?

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

How Can I Subscribe to *ATPM*?

Visit the [subscriptions page](#).

Which Format Is Best for Me?

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

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.

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 strongly we recommend it. Please contact our [reviews editor](#), before you begin writing, if you have a piece of hardware or software that you'd like to review.

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. Products rated Good or better are ones that we recommend. Okay products get the job done. We recommend avoiding Rotten products.

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.

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.

