



Washington Apple Pi

Journal

**Windows
on a Mac
in 2013**





Washington Apple Pi Journal

Summer 2013
Volume 35, No. 2

Copyright Notice

© COPYRIGHT 2013, by Washington Apple Pi, Ltd.
Anyone wishing to reprint material from this publication must first obtain permission. Such requests may be sent by email to **maceditor@wap.org** or by postal mail to the Pi office care of *Washington Apple Pi Journal* Editor. When reprinting any portion of the contents herein, proper author, title, and publication credits must be given. A copy of the article as printed must be sent to: Washington Apple Pi, P.O.Box 6800, Silver Spring, MD 20916-6800.

Contacting Washington Apple Pi

Washington Apple Pi, Ltd.
P.O. Box 6800
Silver Spring, MD 20916-6800
Business Office: 301/984-0300 [message]

Web address: <http://www.wap.org>

Email address: office@wap.org

President Diana King
president@wap.org

Treasurer Raju Tonapi
treasurer@wap.org

Secretary Jed Sorokin-Altman
secretary@wap.org

Directors Jonathan Bernstein
jon.bernstein@wap.org
Jay Castillo
joseph.castillo@wap.org
Ken Goldman
kenneth.goldman@wap.org
Robert Huttinger
r.huttinger@wap.org
Larry Kerschberg
l.kerschberg@wap.org
Jed Sorokin-Altman
jed.s@wap.org
Richard Orlin
richard.orlin@wap.org
Dick Nugent
dick.nugent@wap.org
Jason Woo
jason.woo@wap.org

Managing Editor Jay Castillo
editor@wap.org

Review Editor Lawrence I. Charters
maceditor@wap.org

Senior Copy Editor Patsy Chick
patricia.chick@wap.org

Copy Editor Diana King
diana.king@wap.org

Design & Production Nora Korc
nora.korc@wap.org

Contents

President's Page

Diana King..... 4

Editor's Page

Jay Castillo..... 5

Comic Pi..... 6

Hot Links..... 7

Windows on a Mac in 2013

Lawrence I. Charters..... 8

QR Codes: The Good, the Bad, and the Ugly

Jonathan Bernstein 14

More Hot Links..... 17

How I Got a New 27-inch iMac for \$350

Richard Orlin..... 18

Replacing the Bad Hard Drive on My iMac

Neil Ferguson..... 20

Cryptogram: A Word Puzzle

Jay Castillo..... 22

Implementing LastPass – No Small Endeavor

Bob Whitesel..... 24

FlippedBITS: IMAP Misconceptions

Joe Kissell..... 27

What version of OS X should I run?

Jimmy Obomsawin 31

FlippedBITS: Java, JavaScript, and You

Joe Kissell..... 32

The New Improved Pi Calendar..... 37

Upcoming Washington Apple Pi Meeting..... 38

Members Helping Members..... 39

President's Page

Diana King

By the time you read this, I shall have completed one year as president of this organization and, astonishingly, been reelected. It was a position I never expected to have and still wonder if I am qualified for. Past presidents like Pat Fauquet and Bob Jarecke and Jay Castillo have far greater technical knowledge than I do. There have been some bumps in the road, such as storms that knocked our servers off-line for a few days, and another technical glitch that delayed the posting of some podcasts, but the club has survived the year, thanks to a dedicated cadre of Pi activists who believe in our mission.

Also by the time you read this, Apple will have announced their latest updates/upgrades and new products (*iRadio*, anyone?) at the June World Wide Developers Conference. To keep up to date on their offerings, we intend to invite Apple's Aaron Davis back for a General Meeting as soon as possible. We are also still working on a return visit by Rob Pegoraro and on other programs that members have told us they would like to see.

Our club has approximately 500 members, most of whom live around the DC area, but some of whom have joined or maintained their memberships from afar. I want to thank all of those who loyally renew every year; your dues and donations keep us going. We exist for your benefit and hope that you think you are getting the help and information you need. If you have any ideas for ways to bring in more members, please share those with us at president.wap.org.



Editor's Page

Jay Castillo

It's summertime, and here we are with the summer *Pi Journal*. Our cover article features our longtime Apple/Mac guru, Lawrence I. Charters. Lawrence relates the history of using Windows operating systems on a Mac, and compares the applications *Parallels Desktop* and *VMware Fusion*, which allow you to do this. So make your choice. Jonathan Bernstein explains to us what QR codes are, which apps to use to read them and to make them. So have fun with QR codes.

Need a new iMac, but can't quite afford it? Richard Orlin shows how to get greatly improved performance using an external solid-state drive and a Thunderbolt connection. Save some money!

Neil Ferguson finishes the saga of his iMac's failed internal hard drive with the story of how he replaced it using his very own hands. I provide the first installment of our new Puzzle and Game corner with a review of the word puzzle, *Cryptogram*. Great fun! Please send me reviews of your favorite game or puzzle, even *Angry Birds*.

In a previous issue, Dale Hrabak reviewed the application, *LastPass*. Bob Whitesel relates his experience with installing and using it. Our two reprints are about understanding the IMAP mail protocol, and unraveling Java and JavaScript for Macs. Jimmy Obomsawin gives us a graphic that shows the relative merits of the recent Mac big cat operating systems.

Now we get ready for the fall issue, so I eagerly await your contributions. It's never too soon. Thanks!



THAT EXPLAINS II ...

ill, illness, to fall ill, disease, to be ill,
to become ill

Chinese character (modern = simplified) and **stroke order animation** :



Character: Chinese Simplified
Pronunciation: Hanyu Pinyin
(Mandarin = Standard Chinese)



病

bìng

© HanTrainerPro.com



Hot Links

- Poor iPhone battery life? Make it all better.
<http://www.zdnet.com/15-ways-to-improve-your-iphones-battery-life-7000014902/>
- If you want to know more about using *Dropbox*, there's a new Take Control book for it.
<http://tidbits.com/article/13723>
- Got 10,000 iPhones you don't know what to do with? Make an awesome domino sequence.
<http://floost.com/mactrast-post-watch-the-coolest-domino-sequence-ever-using-iphones-2506884>
- Booklovers, *Delicious Library 3* is here to help you better manage your library.
<http://arstechnica.com/apple/2013/04/delicious-library-3-is-here-and-its-a-beast/>
- Enhance Apple *Mail*'s spam filtering capabilities.
http://www.macworld.com/article/2033767/dealing-with-junk-mail.html-tk.nl_mwhelp
- If you want to find and copy a file path in OS X, here's how.
http://reviews.cnet.com/8301-13727_7-57578129-263/how-to-copy-a-file-path-in-os-x/
- Get severe weather updates on your iPhone regardless of your service provider.
<http://www.insanely-great.com/news.php?id=12926>
- iPads in space! A great music video from Commander Chris Hadfield.
<http://www.cultofmac.com/227268/space-station-commander-chris-hadfield-filmed-edited-space-oddy-using-garageband-on-his-ipad/>

Windows on a Mac in 2013

© Lawrence I. Charters

In the beginning (1976 that is) there was Apple. The IBM PC didn't come along until 1981, and the Macintosh didn't appear until 1984. Windows, a latecomer, didn't make an appearance until November 1985, and wasn't really much of a commercial force until Windows 3.0 was introduced in 1990. But starting with Windows 3.0, the Mac world started to take notice.

Sort of, in a vague way. In 1991, in response to growing consumer interest, Apple released an Apple IIe card, allowing the Macintosh LC family to run Apple IIe programs on a Mac. By 1994, Apple decided to try their previous trick with something other than an Apple operating system and introduced the Power Macintosh 6100 DOS Compatible. This awkwardly named machine was a PowerPC-based Macintosh with an 80486 processor card capable of running MS-DOS and Windows 3.1 software, side by side with the Mac's native System 7 or Mac OS 8 or OS 9.

A few years later, in 1996, there was significant consumer demand for at least some sort of Windows compatibility on the Mac, and Insignia Solutions started marketing a less costly software emulation of a PC computer, *SoftPC*. This was a very impressive technical trick, as *SoftPC* was essentially a PowerPC-based program pretending to be an Intel-based computer, allowing the user to supply their own copy of Windows to install and run Microsoft Windows-based software. Insignia later bundled copies of Windows 3.0, Windows 95 and

Windows 98 with their emulator. The result was slow—one might even say painfully slow—but you could, in fact, run Windows software directly on the Mac, with no expensive plug-in hardware.

Then in 2006, Apple introduced Intel-based Macs. This new architecture was vastly different; Macs and Windows computers were now, for the first time, using the same kind of central processing unit (CPU). Apple exploited this directly with *Boot Camp*, a methodology in which the boot hard drive is partitioned with a Windows-friendly volume that can then be set up with a copy of Windows. While still used, this method has fallen out of favor as, first, Parallels, Inc. introduced a software-based virtual PC environment and then, a bit later, VMware joined the Mac market with their own virtual environment.

By 2013, running Windows on a Mac using either *Parallels Desktop* (hereafter "*Parallels*") or *VMware Fusion* (hereafter "*Fusion*") has become so commonplace that the only real question left seems to be which one is better. *MacTech magazine*, a magazine aimed at Mac developers, network managers, and hard-core Mac wonks, decided to answer that question with a 9,200-word report. They gave *Washington Apple Pi Journal* permission to mine their article mercilessly as long as we promised to mention their MacTech BootCamp conferences, which they hold around the country. See links to the original MacTech study and the MacTech BootCamp schedule under Resources at the end of this article.

Virtual Benchmarking

When Parallels, Inc. first introduced their virtual environment, called *Parallels Workstation*, in 2006, virtual machines on the Mac seemed simple: Mac users wanted to run Windows, specifically, Windows XP. It was possible, if you wanted to be strange, to install and run Windows 3.1 or 95 or some other flavor, but Windows XP was the most sought after.

A year later, when *Fusion* stepped in to offer competition, the world was starting to get more complicated. By this time, running Windows XP seemed commonplace, and Mac users were experimenting with running Windows Server 2003, Windows Vista, and various flavors of Linux. Parallels, Inc. and VMware found themselves in a battle to see who could claim to offer the easiest, smoothest, most “Mac-like” experience of running decidedly Mac-unlike operating systems. This battle has grown more complex over the years with the introduction of Windows 7 and, most recently, Windows 8.

MacTech decided to deal with this complexity by trying pretty much everything. They tested both *Parallels Desktop 8.0* and *VMware Fusion 5.0* (the latest versions of each) with both Windows 7 and 8.

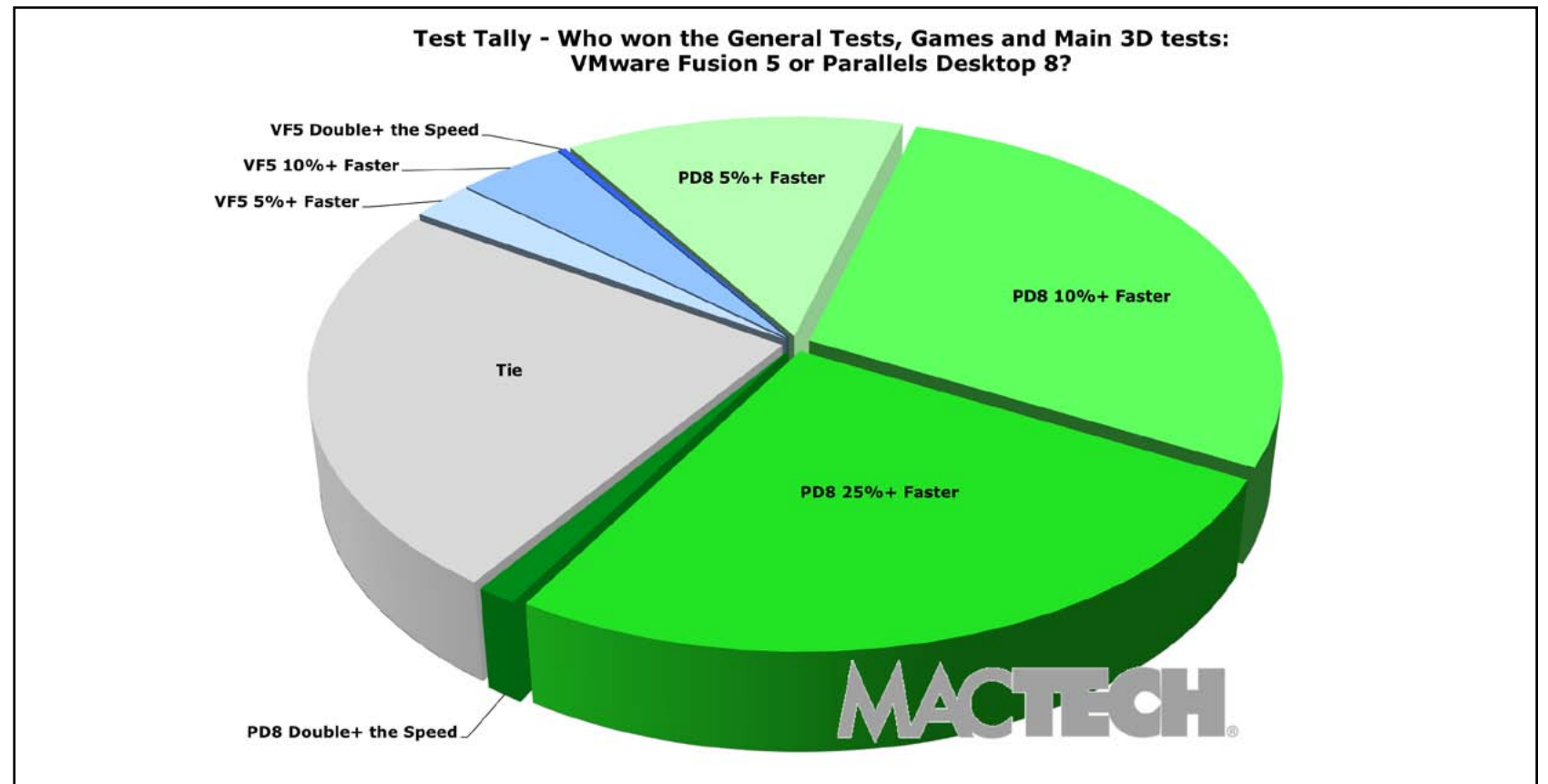


Figure 1: *Parallels* won the vast majority of the general tests, games tests, and tests of 3D graphics. In about a quarter of the tests, *Parallels* and *Fusion* tied, and a tiny sliver of the tests were won by *Fusion*.

They tried running different versions of Mac OS X under each (yes, a Mac running Mac OS X running another copy of Mac OS X in a virtual environment). They also tested a wide range of hardware: a 1.8 GHz MacBook Air 13-inch dual-core i5 processor, a 2.3 GHz MacBook Pro 15.4-inch quad-core i7 processor, a 2.7 GHz MacBook Pro with Retina 15-inch quad-core i7 processor, a 2.7GHz iMac 27-inch quad-core i5 processor, and a 3.2 GHz quad-core Mac Pro. All the Macs were running OS X 10.8.2.

Need for speed

Technically, MacTech’s tests are classed as “benchmarking.” These tests were mostly automated tests designed to show differences in speed, or capacity, or some other quantifiable measure. The tests are not aimed at determining how easy it is to use *Parallels* or how the configuration options of *Fusion* compare or which company has the best tech support or anything else dealing with day-to-day usability. Using such benchmarks, *Parallels* came out the clear winner (see Figure 1).

MacTech found the results so startling that they released a color-coded spreadsheet of the results, with *Parallels* wins marked in green and *Fusion* wins marked in blue; ties and tests not performed were left uncolored. You can't help but notice the spreadsheet is remarkably green (see Figure 2, previous page).

In all honesty, both packages do a more-than-adequate job of running Windows on any current Mac with a decent amount of memory. If you have a machine more than four years old or with less than 4 GB of memory, performance may be an issue, but for most users, except high-end gamers, things should be fine.

MacTech didn't stop with performance issues; they also discussed "footprint" (how much processing overhead the packages used) and battery tests. Footprint tests showed both *Fusion* and *Parallels* were remarkably civil and courteous packages; while *Parallels* won the tests, either could sit by idly and not interfere with anything else you might be doing. In battery tests, *Parallels* outdistanced *Fusion* in stretching out battery life, a critical consideration if you want your Mac laptop to also serve as your Windows laptop.

Games

For gamers, 3D graphics tests are of keen interest; if you are up to your armpits in orcs, battling your way through an evil forest, you don't want slow graphics to put you at a competitive disadvantage when you are attacked by a grue. MacTech ran a ridiculous number of graphics tests, but they also took the time to simply play games the old-fashioned way, with a person at the keyboard. Here, performance issues were less of a problem than artifacts: flaws appeared on screen, little cracks in the painstakingly rendered universes of the game. In



Figure 3: This screen shot shows *Parallels* playing *Torchlight II*, a fantasy game. A panther (indicated by the arrow) appears as a wireframe rather than a texture-mapped feline carnivore. *Fusion* had the same problem, with the same game.

Figure 3, playing the game *Torchlight II*, both *Fusion* and *Parallels* had problems with a panther appearing as a wireframe rather than a fearsome feline creature. Games, in fact, encompass a large section of the MacTech report, reinforcing the old adage that "Macs are for work, Windows for massively decreasing the productivity of the American workforce." If you want to find out how *Fusion* and *Parallels* handle space marines, Formula 1 racing, and other essentials, refer to the full report. Generally speaking, gamers should consider *Parallels*, a MacBook Pro

with Retina display and lots of memory, and the largest possible solid-state drive. True, almost everything works with a lesser machine, but the editors clearly had the most fun with this combination.

Retina support

One interesting issue is support for Retina display. The Windows world doesn't have anything like the MacBook Pro with Retina display, and not just because this machine has an awkward name. Windows simply doesn't understand the high-density character of the screen, so *Parallels* builds in support to transparently map Windows to something useful. *Fusion* relies on Windows built-in preferences, resulting in tiny screens about one-fourth the desired size. VMWare expects the user to manually change the size of text and other screen elements to 199% in order to make things readable. While this sounds easy to do, it also sounds less than elegant.

The crispness of the Retina display shines in both Windows 7 and 8, on both *Fusion* (when adjusted) and *Parallels*. MacTech has a number of screen shots showing how nicely text and icons appear on the Retina display. On the other hand, *Fusion* does fall down because of its failure to size things appro-

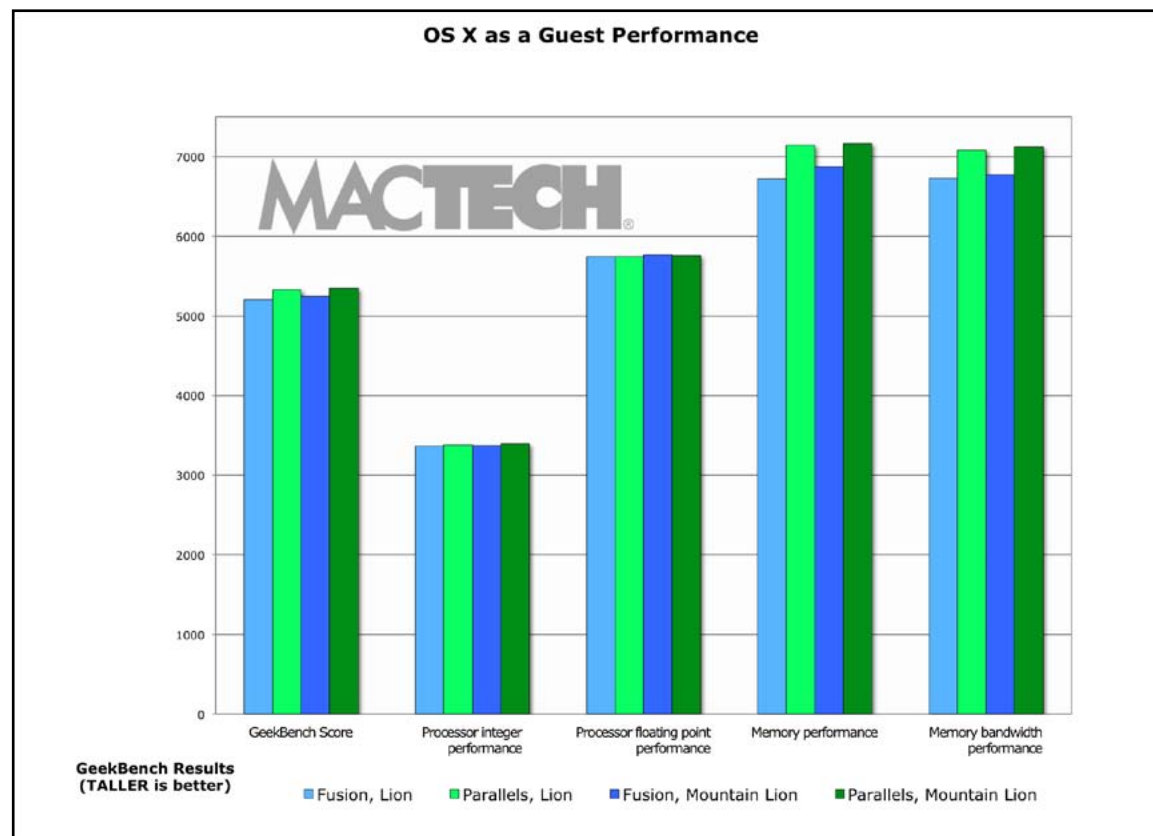


Figure 4: Running Lion and Mountain Lion under both *Fusion* and *Parallels*. As you can see, *Parallels* usually came out on top, but the differences are not that remarkable.

priately under Windows 8. One set of screen shots shows the opening screen in Windows 8 looking quite tiny under *Fusion*, while *Parallels* displays the same as you would expect.

Operating system support

As of this writing, there are few compelling reasons to run Windows 8 instead of Windows 7, but if you absolutely, positively need to run Windows 8 on your Mac, you'll need to use *Fusion* or *Parallels*. *Boot Camp* can't run Windows 8.

Both *Fusion* and *Parallels* have support for running Mac OS X 10.7 and OS X 10.8 as a virtual operating system. MacTech points out that this is not something most people would want to do; why would you run Mac OS X as a guest operating system of Mac OS X? But you can, so they tested both packages, and both performed adequately (see Figure 4). They suggest that this trick is probably of most use to developers, system administrators, and other strange creatures, testing out software builds or odd configurations.

Personal observations

I've used various flavors of Windows on Macs since the days of *SoftPC* and have used *Parallels* since it was first released. For reasons that I won't bother to explain, I regularly run Windows XP, Windows 7, Windows 8, Linux, and Chrome OS under *Parallels*, and have no particular problems with any of them. I've never tried to run Mac OS X under *Parallels*, as the installation process is a bit cumbersome and there is no good reason for me to try. MacTech didn't cover Linux or Chrome OS, but I'm reasonably confident *Fusion* could handle both.

One topic not covered in the MacTech article is the "disposable" nature of a virtual machine. Both *Fusion* and *Parallels* allow you to duplicate installed virtual machines, and this is a terrifically powerful tool for experimentation. Afraid the latest update patches may "break" something in Windows? Make a duplicate of your installation, and then install the patches on the copy. If it works, fine. If it doesn't work, throw the copy in the trash and empty the trash; your original, untouched installation is still perfectly usable.

Trust me: thinking of Windows as disposable makes it much easier to put up with. And fun, as you can simply flush your frustrations away.

If you've ever wondered which was better, *Fusion* or *Parallels*, read the full MacTech article. If you've ever wondered how many ways you can torment an operating system, read the MacTech article. In particular, if you are a gamer and like playing Windows-based games, but would rather use your Mac, check out the MacTech article. It is as insightful as it is geeky.



Resources:

Original MacTech Virtualization Benchmarks article:

<http://www.mactech.com/2013-01/virtualization-benchmarks>

MacTech BootCamp II conference description:

<http://www.mactech.com/bootcamp/about>

MacTech BootCamp II conference schedule and locations:

<http://www.mactech.com/events/locations>

QR Codes: The Good, The Bad, And The Ugly

© 2013 Jonathan Bernstein

If you have wondered about the square barcodes that seem to be popping up everywhere, read on.

Background

Businesses have been using barcodes for many years. We are all familiar with the traditional barcodes that look like a block of vertical lines and that get scanned whenever we buy something at the grocery store. Quick Response (QR) Codes are a newer form of barcode that can be read by a cell phone. They are two-dimensional, and can store much more data than a standard barcode, including URL links, geo coordinates, and text.

As an example, the Silver Spring Regional Service Center in downtown Silver Spring used to put out a two-page publication describing its mission, services, and resources for the community. Then an Americorps intern at the center who knew about QR Codes changed the center's publication to a small card, roughly three inches by four inches, with a QR code. Scan that little square and you can get all the information that was in the old two-page publication (Figure 1).



Figure 1: Welcome to Silver Spring card—9 to a sheet.

Let's explore where these QR codes can be found, how to read them, and how Pi members can create and use them for a range of purposes.

Where to Find QR Codes

Pay attention when you walk around, and you will get an idea of how ubiquitous these QR codes are. Once I started looking, I was surprised how often I noticed them. For example, I ran across a QR code on a publicity poster about an arts festival at The Atlas Performing Arts Center in Northeast Washington, DC (Figure 2). Then I noticed a QR code

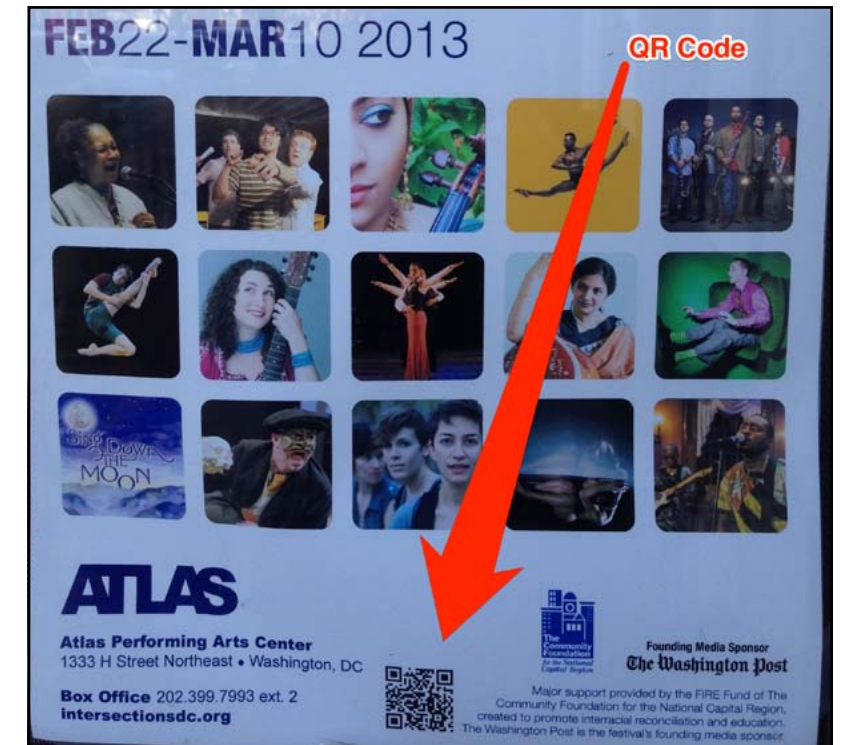


Figure 2: Atlas Performing Arts poster.

on the window of a high fashion store in Madrid (Figure 3).

On a drive to New England to visit my family, at a turnpike rest stop, I saw a travel sign that contained QR codes for things ranging from tourism information to hotel information, to travel advisories, and even webcams for traffic (Figure 4). Face it folks, when something comes to the New Jersey Turnpike, we should all know about it!

How to Access QR Codes That You Come Across

To read a QR code using your iPhone or iPad, you'll need an app. I've used three thus far. But there are dozens of them out there; this discussion is illustrative, not exhaustive.

The first app I used, called *Scanlife*, is a simple free app for scanning QR codes. It opens to a scanning window, and when one scans a QR code with the iPhone camera, it makes a satisfying sound. If it detects a URL, it will ask whether the user wants to open a website, and then quickly does so in a new window, with sharing options that include opening in the full *Safari* browser. It also allows you to generate a barcode for your contact info and saves it to the camera roll for easy display to other QR scan users. But it cannot otherwise generate QR codes, and it comes with annoying ads that run along the bottom.

I also used *QR Reader for iPhone*. This free app (with ads) is more full featured. Its scanner similarly reads QR codes, opens URLs in a window without asking, and has an option of opening in the IOS browser. It also allows a user to generate QR codes for a range of purposes (discussed more below).



Figure 3: Madrid store window.

Last, I've also used *Bakodo*, one of the many shopping apps that started reading product barcodes to compare prices (like *Shopsavvy* and its ilk) and added QR code scanning to its functions.

In my brief use of these apps, for just the raw scanning and viewing of QR codes, I found *Scanlife* to be the best. *QR Reader for iPhone*, sometimes took

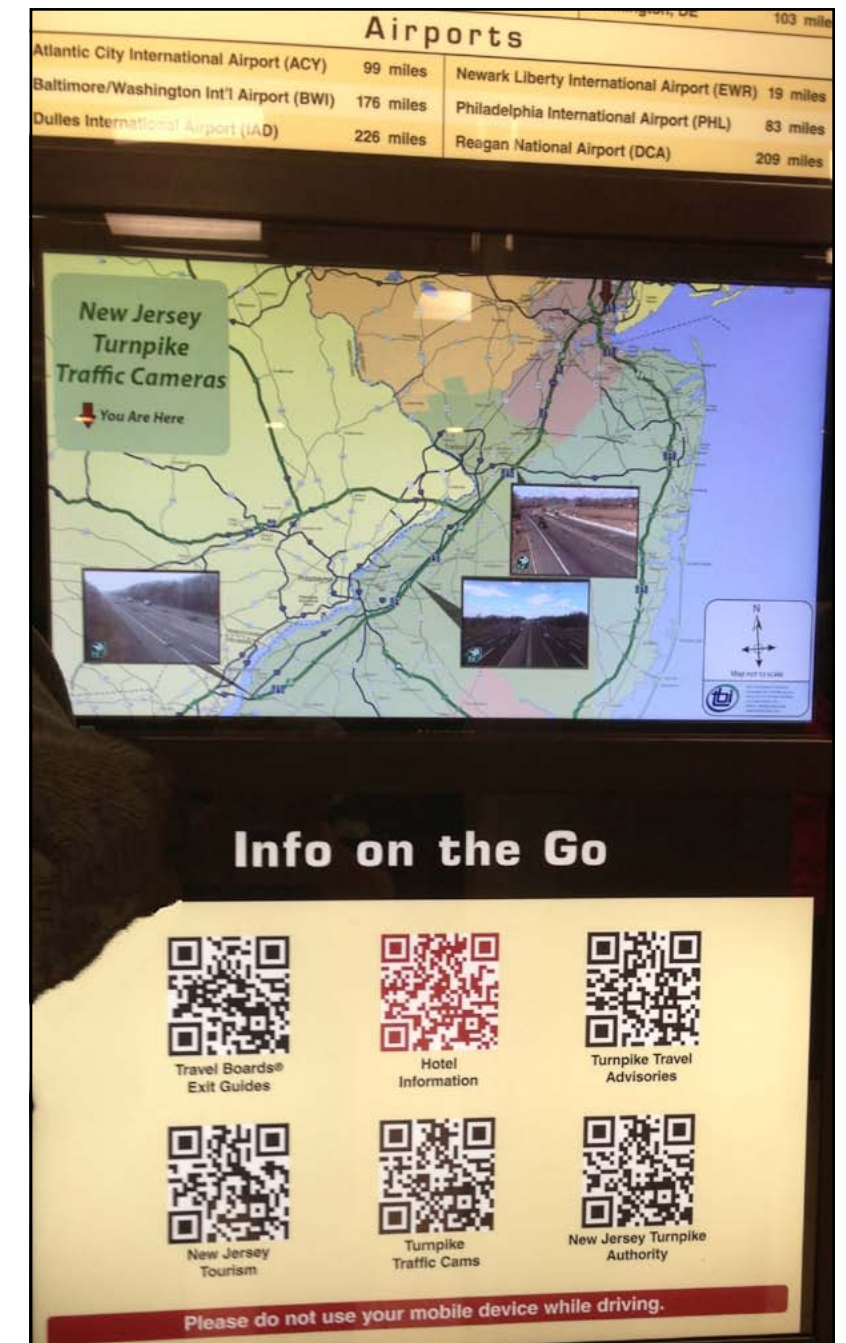


Figure 4: New Jersey Turnpike information



Figure 5: qrstuff.com options for generating QR codes.

longer to read QR codes, and it moves less easily back to scanning mode from the page displaying the scanned contents.

How to Generate QR Codes. Generating QR codes is easy, either by using an iOS app like *QR Reader for iPhone* or by visiting one of the many websites that freely allow users to generate and download a QR code. One of them, qrstuff.com, does not even require registration for an account. What is remarkable is the range of data that can

be encoded, from URLs to websites, text strings, map locations, email addresses, the Pay Pal “Buy Now” link, and many others.

Figure 5 is a screenshot of the data types that qrstuff can generate. The app *QR Reader* is more colorful with its list of data types that it can generate in a QR code (Figure 6). In addition to generating QR codes that open URLs, I also captured an extended quote from Pi Treasurer Raju Tonapi’s recent piece about getting help from the Pi. Good quote, Raju! (Figure 7).



Figure 6: QR Reader options for generating QR codes.



Figure 7: Text quote from *Pi Journal*.

Among the other websites that allow free generation of QR codes, some will generate only URLs or text. One example of a paid full-featured site used by business marketers is [Visualead](#).

Conclusion

I will probably generate a QR code for Wi-Fi details, to make it easier for Pi members to access our Wi-Fi at the Cedar Lane space or at our GMU meeting space. So put on thinking caps about how QR codes can help amplify messages in activities ranging from business efforts to personal activities, and send feedback to this article.

More Hot Links

- Be smarter about your Mac basics. Here are terms you should know:
<http://www.macworld.com/article/2043462/20-computer-terms-every-mac-user-should-know.html> - tk.rss_all
- What time is it? Are you hoping for an iWatch? Check out these speculations:
<http://tidbits.com/article/13892>
- Sure, you want to know everything. Here is a user friendly technical dictionary:
<http://pogue.blogs.nytimes.com/2013/06/20/a-technical-dictionary-that-fits-the-definition-of-user-friendly/>
- The new Apple Airport Extreme and Time Capsule introduce the 802.11ac next generation of Wi-Fi:
<http://www.wired.com/gadgetlab/2013/06/802-11ac-apple-wwdc/>
- If you really want to do it yourself, iFixit provides repair manuals, parts and tools for Macs, iPods, iPhones and iPads:
<http://www.ifixit.com>



How I Got a New 27-inch iMac for \$350

Richard Orlin

Before you turn the page thinking that this must be some kind of scam, please let me explain. It wasn't really a new iMac; it just performed like one.

For the last several months, my two-year-old mid-2011 iMac seemed to be running more and more slowly. Apps would take a long time to start. Their icons would keep bouncing in the dock, taunting me. Spinning beachballs would haunt me whenever I tried to do something that required disk access. I cleaned out all unused programs and files that had accumulated in the last two years. I ran Disk Utility and repaired permissions and repaired the disk, all to no avail. I even tried Disk Warrior, but that didn't ease the pain. As a last resort, I nuked the drive and rebuilt the system from scratch. The end result was not much better than before. To make matters worse, my new Macbook Pro has an SSD drive, so the iMac seemed even slower by comparison. I was on the verge of throwing up my hands and buying a new iMac, but then I thought, having retired several months earlier, I should not be spending almost \$2000 unless it was absolutely necessary.

A better idea might be to replace my hard drive with an SSD drive. I started to research this possibility at Other World

Computing, http://eshop.macsales.com/shop/turnkey/iMac_2011/27_Inch — where I found that it would be cost-prohibitive (\$520). I also found that I would probably be without the iMac for a week, since I wasn't going to attempt the job myself.

Then on a recent trip to Microcenter in Fairfax, VA, their Mac expert suggested an external SSD drive connected to the iMac via Thunderbolt. He showed me two external Thunderbolt SSD drives, but my jaw dropped when I saw the prices: almost \$400 for 128Gb and almost \$600 for 265Gb. Back I went to the Internet for more research. I remembered seeing a Seagate GoFlex sled with a Thunderbolt connection for under \$100 (<http://tinyurl.com/c8ozprd>). The GoFlex system is designed to provide users with a base hard drive that can easily be upgraded with different add-ons. Further research told me that the Thunderbolt sled would accept any brand of portable drive as long as it had a SATA connection (Figure 1).

The next purchase was the SSD drive. Since this would be running 24/7, I wanted a drive that was reliable. I checked a LOT of reviews and came away with only one recommendation: Samsung 840 Pro 256Gb SSD (<http://tinyurl.com/cyqw8s>). All the reviews said this was a stellar product. Samsung claims



Figure 1: Empty Seagate Backup plus Thunderbolt adapter.

As a last resort, I nuked the drive and rebuilt the system from scratch. The end result was not much better than before.

a mean time between failures (MTBF) of 1.5 million hours and a 1500G-shock resistance, and the 840 Pro is backed by a limited five-year warranty. Well, I'll be long dead before the drive reaches 1.5 million hours. Besides, I will most likely replace the iMac before the five-year warranty is up.

The Seagate sled was \$97.88, the drive was \$228.99 and the Thunderbolt cable was a piddling \$29 for a total of \$355.87 from Amazon. Prices on Amazon fluctuate, so it pays to shop around.

Pricing and purchasing were the most difficult parts of this project. After the equipment arrived, the iMac made the installation itself a breeze. The drive easily fit onto the sled. I didn't think it would come loose, but I secured it with some rubber bands anyway, because I'm a belt-and-suspenders kind of guy (Figure 2).

I plugged one end of the Thunderbolt cable into the sled and the other end into the Thunderbolt port on the Mac. The drive is self-powered by the Thunderbolt port, so no worries there. Then I opened up Disk Utility and initialized the drive. (Instructions on the use of Disk Utility can be found at



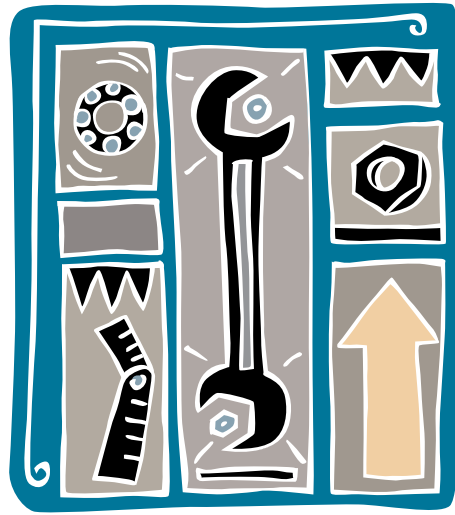
Figure 2: Samsung 840 Pro SSD installed on Thunderbolt adapter.

<http://www.iclarified.com/entry/index.php?enid=1075>.)

If you choose to do this upgrade, you should feel comfortable working with Disk Utility and Migration Assistant. If you are not comfortable working with those two utilities, then this upgrade is not for you. Once your drive is formatted for use, you can install a clean copy of OS X on the drive by using your recovery partition (Lion and Mountain Lion) or your system disk (for earlier operating systems). If you are installing Lion or Mountain Lion, your Mac should automatically reboot and you will be asked if you want to transfer accounts and settings. Here's the caveat: if you have a user folder substantially smaller than your new SSD, it makes sense to transfer both account and applications to your new drive. I think it would be safe to do so if your user

folder is half the size of the new drive. If not, then say no to the transfer and just set up a new account on the new drive using your current username. Then run Migration Assistant, found in the Utilities folder, to transfer only apps and network settings to the new drive. Once Migration Assistant is finished transferring the files, reboot your computer. You will be amazed at how fast it boots. My iMac went from two and a half minutes down to twelve seconds!

If you have only transferred apps, there's just a little cleanup to be done. You have to set pointers to all the data that your apps write to in your home folder: downloads in *Safari*, *iPhoto*, *Word*, the Documents folder, *iTunes*, etc. If you are using for backup, you have to make changes there also. Basically, for any program that saves data in your home folder, you'll have to open its preferences and point to your user folder on your internal drive rather than the SSD. The work and time to get it right are worth the effort. Of course, if you have a small home folder to begin with, there is nothing to clean up, since you already copied everything over using Migration Assistant. Now you're finished; go ahead and enjoy your "new" Mac.



Hardware Tutorial

Replacing the Bad Hard Drive on My iMac

Neil Ferguson

In the last *Pi Journal* issue, I wrote about my problems with the failed hard drive on my iMac, and promised to follow up with a report on my adventures as I tried to replace it. I started of course with research. I read that iMac internal hard drives have a temperature sensor that regulates the speed of the computer's fans. Some commenters said that after installing a new hard drive, the computer's fans would suddenly begin running at full speed. Others said that they didn't hear any fans running at all and that upon checking the internal temperature they saw zero degrees, suggesting that the fans weren't running. I also read that the replacement hard drive must be the same kind as the original. My internal hard drive is a Western Digital 500 GB. Checking its model number on Western Digital's website, I found that it was a Western Digital Blue (whatever that means). The 1 TB external hard drive that I had been using was also a Western Digital Blue. I could find no information on Western Digital's website about any temperature sensor built into either hard drive. What to do now?

At the Apple store I spoke with someone at the Genius Bar. The only thing he could tell me is that the computer needs to be connected to the temperature sensor in the hard drive, and

that if my hard drive didn't have a temperature sensor in it, my computer might not work.

At Micro Center in Rockville, one of their technicians told me that all hard drives manufactured in the past many years have temperature sensors built into them. That was good news. I decided to take the plunge and replace my bad internal hard drive with my 1TB external hard drive

I had thoroughly researched the internet about replacing hard drives in an iMac. YouTube had many videos about opening up iMacs and replacing hard drives and power supplies. There seemed to be a video on every version and size of iMac, and there were numerous articles on the internet about temperature sensors on or in hard drives that Apple uses. I felt I was prepared for whatever I might find once I opened up my iMac. (I like to know ahead of time what is inside something I am going to open up; I don't like the thing to go splannggg and leave me wondering what it was and where it went.)

I began by asking a friend to come over with his tools, including the suction cups needed for opening up an iMac. Not only did I have moral support, I also had a helper who had already

Not only did I have moral support, I also had a helper who had already replaced a hard drive in his iMac. With a bit of trepidation I began opening up the iMac with my friend's tools.

replaced a hard drive in his iMac. With a bit of trepidation I began opening up the iMac with my friend's tools. The glass came off just fine. The frame, after I disconnected the camera, came off easily. I took out the screws holding in the LCD screen and then gently lifted the screen. Three cables connecting the LCD screen to the innards of the computer had to be disconnected. I successfully removed two of them, but the third one looked a bit scary, so I left it in place. I could see the hard drive and I also saw the temperature sensor that was glued to the outside of the hard drive.

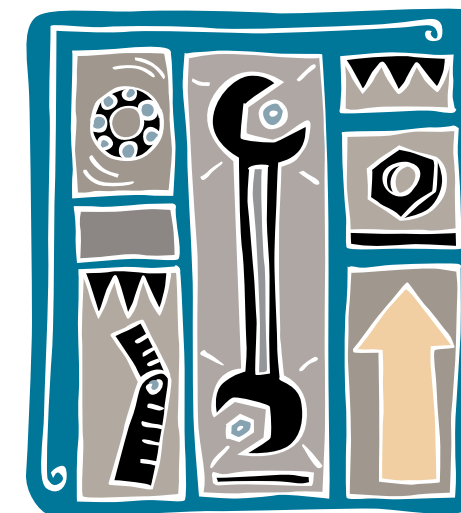
I started to remove the hard drive, but everything that I had seen or read about removing it didn't seem to work. I was supposed to use 'quite a bit of force,' but I didn't really understand about how, where and in what direction to use this 'quite a bit of force,' so I didn't. I did, however, see four screws in the hard drive that I assumed were holding it in place. I successfully removed three of them but was unable to remove the fourth screw. With three of the screws removed I was able to jiggle and move the hard drive. I continued jiggling and moving the hard drive and lifted it up and lo and behold it came out — it didn't need to have the fourth screw removed. In fact, now that I saw how the 'quite a bit of force' was to be applied, I realized that I didn't need to remove any of them. I immediately returned the three screws that I had earlier removed. (After putting in my new hard drive I had fun popping

the hard drive in and out of its seat using 'quite a bit of force.' It works great once you know how.)

I put the hard drive that I had been using externally into the iMac without difficulty, reconnected the SATA and power cables, and used some silicone sealant to fasten the temperature sensor onto the case of the hard drive. I reconnected the two cables to the LCD screen and fastened the screen back into the computer. Finally I put the frame into place, replaced the camera connection, and then fastened the frame back on.

Now for the test: Was the computer going to start up after all this? I connected the keyboard, mouse and power. I closed my eyes and turned the computer on. I heard a bong and was so excited that I opened my eyes and soon saw that little gear wheel spinning. These were good signs. After a little while I saw all the icons of my desktop reappearing. Needless to say, I was and still am a happy camper.

There is one sad part to my tale. On my original hard drive (the one that crashed) there was a partition that held Windows. The external hard drive that I had been using, which fortunately had a Mac OS system on it, did not have a partition with Windows. So, I can no longer start up my iMac in Windows. But who needs Windows anyway, right?



After putting in my new hard drive I had fun popping the hard drive in and out of its seat using 'quite a bit of force.' It works great once you know how.

Cryptogram: A Word Puzzle

Jay Castillo

Cryptogram is an iOS word puzzle in which the words of a quotation are encrypted and presented to the player to solve. Figure 1 displays the opening screen. The Puzzles button allows you to choose the level of difficulty you wish to play; you have a choice of easy, medium, or hard puzzles. The Settings button gives you a number of choices of settings, the most useful of which are to show incorrect guesses in red and to show the frequency of times each letter is used. You can also choose to purchase more puzzles on this screen.

Tapping the Play button produces the current puzzle, as shown in Figure 2. The cryptology incorporated in these puzzles is the simplest type, a mono-alphabet, simple substitution cypher. It is probably produced by randomly drawing a cypher letter for each plaintext letter of the original quotation. As such there is no cryptographic key to find. The most powerful tool for attacking these cyphers is the letter frequency statistics of the cryptogram. That is why selecting the Show Frequency option in the settings is useful. In the

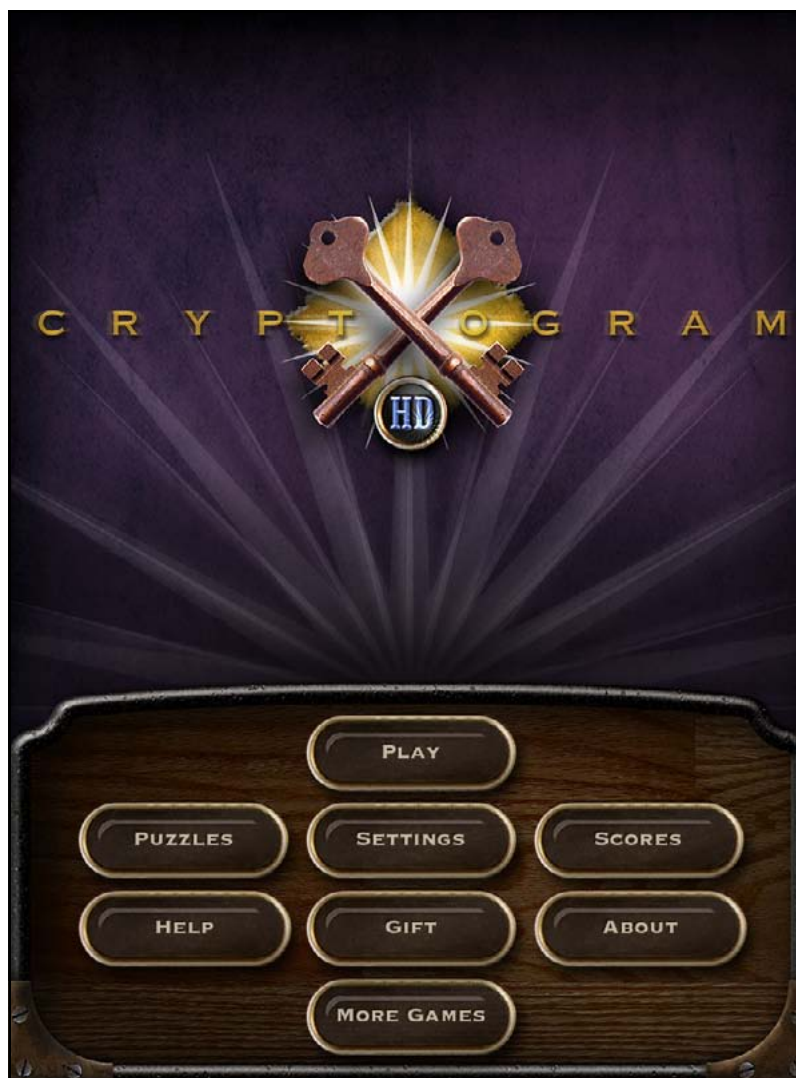


Figure 1: The opening screen for *Cryptogram*.

Figure 2 puzzle, the frequency of occurrence of each letter is shown above its position. The frequency of letter occurrence in the English language is well known:

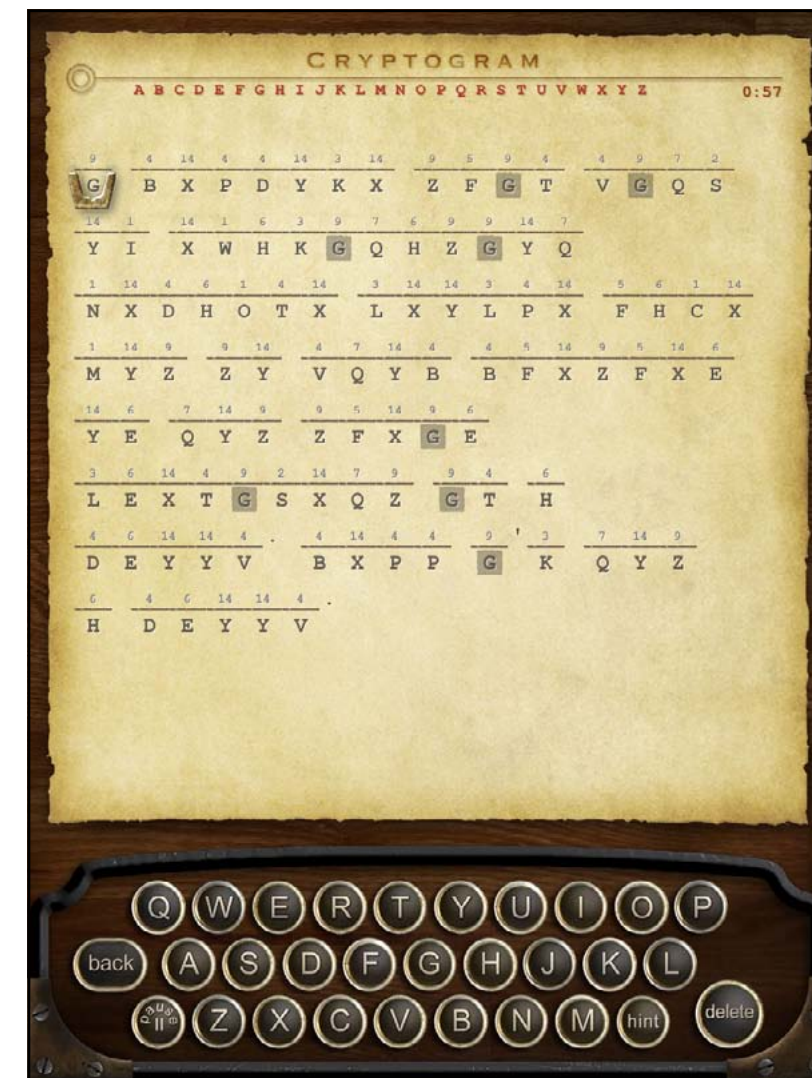


Figure 2: An example of a cryptogram puzzle.

(<http://oxforddictionaries.com/words/what-is-the-frequency-of-the-letters-of-the-alphabet-in-english>). The most frequent letters, in descending order are: E,T,O,I,N,S,H,R,D,L,U. So one can identify the letters occurring most

often in the puzzle and try them in order. In parallel with letter frequency, one can identify any one- or two-letter words, or double-letter words. There are limited letter options for these words. For example, in the Figure 2 puzzle, the fifth word has two letters; so it should be “is,” “it,” “on,” “at,” “be” or one of a few other words. On the second try, the first letter turned out to be “o.” After two more tries, the second letter was “f.” If there is a one-letter word, it can only be “I” or “A.” Once correctly selected, the plaintext letter is shown for each occurrence in the puzzle, and the corresponding letter in the alphabet at the top of the screen is greyed out. The more letters you identify, the easier it is to identify the plaintext words and solve the cryptogram. Figure 3 shows the prize you win by solving the puzzle.

Figure 4 shows a completed puzzle. The upper right-hand corner of the screen displays the time for the solution. The easy puzzles are expected to be solved in three minutes or less; medium puzzles, four minutes or less; and hard puzzles, more than four minutes. Scoring includes penalties for incorrect guesses and hints. You can check your score by using the Scores button shown in Figure 1.



Figure 3: The award for completing a puzzle.

I don't care about my score, so I just focus on getting a solution without hurrying. Touching Completed in the Puzzles button in Figure 1 provides the author of each completed puzzle quotation, arranged by puzzle difficulty.

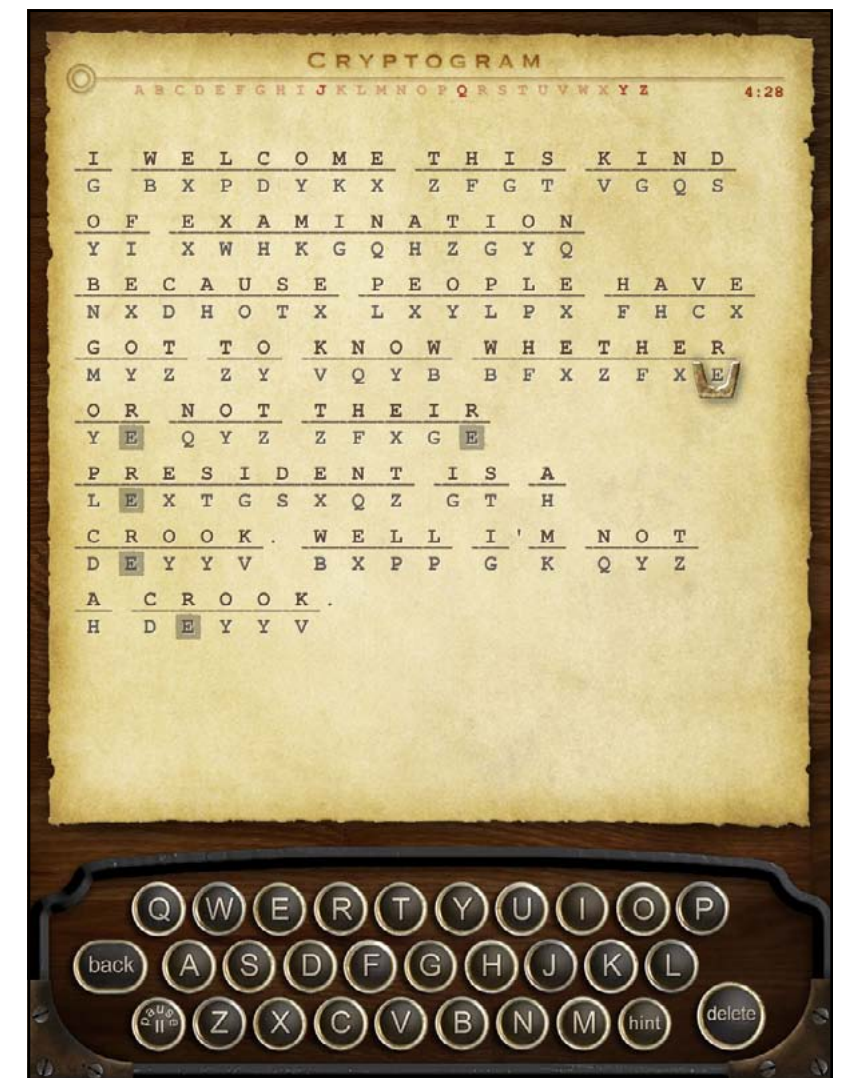


Figure 4: A completed puzzle.

Cryptogram is a simple, fun word puzzle that can be quite challenging, especially when the cryptograms have a flat letter frequency distribution. It's good mental exercise and doesn't require you to be online to play. It's available in the App Store for \$1.99.

Implementing *LastPass* – No Small Endeavor

Bob Whitesel

After I read the articles by Dale Hrabak and Jonathan Bernstein on *LastPass* and *1Password* in the November–December 2012 issue of the *Pi Journal*, I got inspired. I had heard a number of positive comments about *1Password*, and I suspect it's a bit more Mac-friendly, but I missed out on the half-price deal Diana King alerted us to in February. So, I re-read Dale's article and downloaded *LastPass*.

This article describes my experience implementing *LastPass*. If you are seriously thinking about giving *LastPass* a try, I recommend reading Dale's overview article first. My article is about the nuts and bolts of this app. Implementing *LastPass* has turned out to be a very time-consuming task. Nevertheless, my early experience has been positive.

Getting Started

For starters the *LastPass* folks recommend that Mac Users first download the *Firefox* version (www.lastpass.com) and use it to create entries in your *LastPass* vault. The transition to Safari can come later.

After you have installed the *LastPass* plugin into *Firefox*, the first thing to do is to construct a really secure, easily remembered Master Password. This will be your access to your vault. No one except you knows this password — *LastPass* Help says they cannot retrieve it for you — so it is especially important that it be easy for you to remember.

Once I had created the Master Password, I began visiting my sites and using my current list of USERID/password combinations to gain access. Then, I proceeded to find the page at the site where I could change my password. Most such pages will ask for your old password, then give you two dialog boxes to (1) enter a new password, and (2) confirm it. *LastPass* will detect that step and show you a scarlet banner under the toolbar. It will ask if you'd like *LastPass* to generate a new, secure password for you. Click on GENERATE and *LastPass* will show 12 random letters and numbers. I had *LastPass* generate the passwords because some of my sites accept only letters and numbers. In Dale's article he shows a

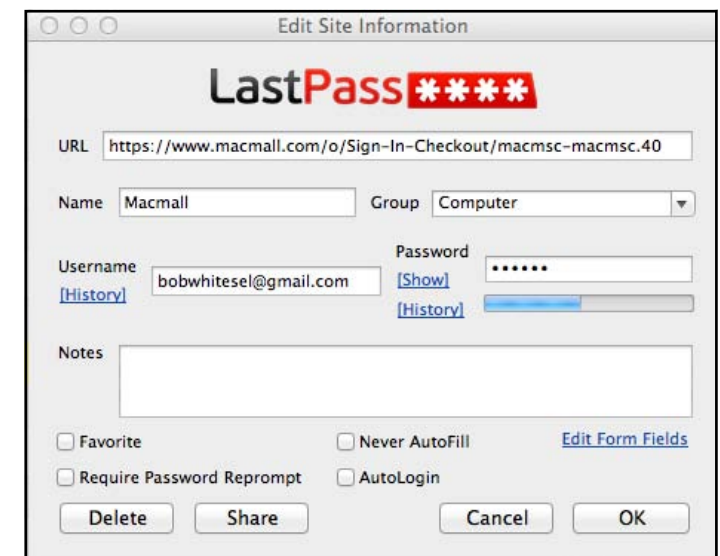


Figure 1: *LastPass* dialog box displaying data for my MacMall site.

screen shot of the advanced options, including the option for special characters.

If you click GENERATE, then ACCEPT, *LastPass* will put the 12-character password it generated into both dialog boxes. Then, you can click SAVE (or whatever the site page displays as your next step) to save the new password AT THE SITE. *LastPass* will detect this action and ask you to confirm your action. Then, it will display a dialog box like the one shown in Figure 1 that allows you to save the password to the *LastPass* vault.

As you can see from Figure 1, I not only saved my login to MacMall, I also edited the NAME dialog box to show MACMALL and assigned it to a group named COMPUTER. After doing a first site, I recommend that you stop and construct a list of appropriate group names so you can place sites as you go (see Figure 2). Use the CREATE GROUP button on the ACTIONS menu to add groups.

At present my BANK group is empty for the same reason Dale discussed. Until I'm confident in this system, I won't automate login to sensitive sites.

Two items to note: First, *LastPass* rates the security of my passwords at just 38%. When I first started, I was batting only 17%. The biggest reason is that I have been using the same password for multiple sites. This is a poor practice and I need to correct it. Second, one of my groups is named SECURE NOTES. I think this works in a similar fashion to Secure Notes under *Keychain*. I have found this feature more important as I put more information into my machine. For example, while adding the site of the local natural gas company, things went haywire. All of a sudden I needed our account number at the gas company to communicate with a representative on the phone. Copies of paper bills are long gone.

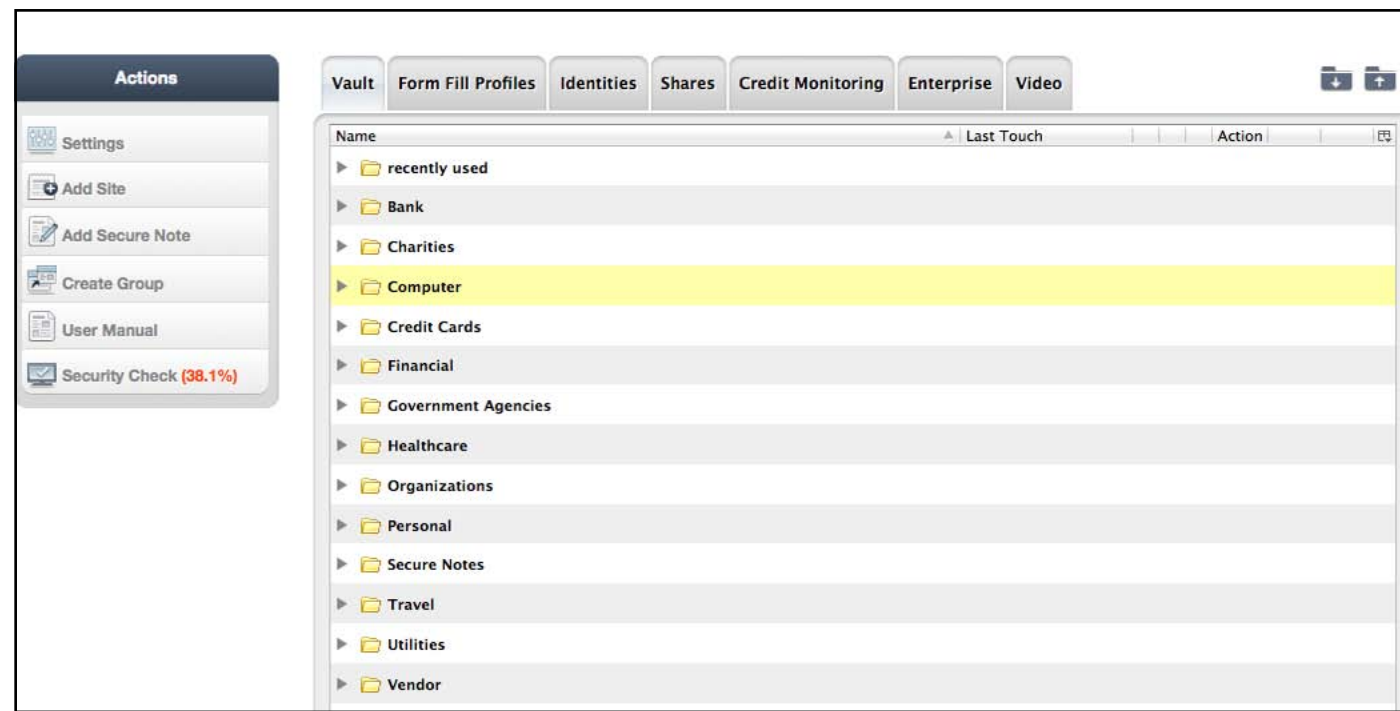


Figure 2: *LastPass* vault showing the groups chosen to organize my sites.

What to do? Eventually, I found the account number and succeeded in adding the gas company site to my vault. Now, I have that account information in Secure Notes.

What if your site requires more information?

As long as your site requires ONLY (1) a UserID and (2) a password, this process of adding sites goes fairly smoothly. Now, suppose you have a site that requires both of the above plus a PIN, or has security questions and answers required for login. Time for an alternate tactic. After you log in to *LastPass* with your e-mail address and Master Password, a small box in the upper right-hand corner of the

screen will turn into a red box with a white asterisk in the middle. This is your access to an important drop-down menu, shown in Figure 3, that you will need for these sites.

Go to the site's CHANGE PASSWORD dialog box as before. Follow through with all the PINs, security questions, etc. Let *LastPass* generate a secure password if you choose. Then, BEFORE you submit all this to your site, click on the DROPDOWN menu next to the white asterisk. Scroll down to SAVE ALL ENTERED DATA and click. After that, you can click to send data to the site's host, confirm the entries with *LastPass*, and see what happens. The *LastPass*

video tutorial that describes this process and can be found at: https://www.youtube.com/watch?feature=player_embedded&v=LmYQM0bhNg4. This procedure does work but may take some work to get *LastPass* to play nicely with each of these sites that require more than simply a UserID and password.

Checking your work as you go

On the menu in Figure 3, scroll down to Tools>Print>Sites. This gave me a very tight little table in small font that I could not edit with any app on my Mac, that contained an alphabetized list of site names, group assignment, URL, Username and PW. Actually, keeping a list like this around could sabotage a lot of your work, so it is best shredded when you're confident of the contents of your vault. For me it helped to find errors. As an example, our AT&T wireless phone website has insisted on using a cell phone number as a UserID, a poor practice. Now, AT&T lets you define an "AT&T Access ID," which is preferably an e-mail address instead. In the printed table I discovered the access ID was incorrect and went back to fix it.

Now, what are my next steps?

I have now more than 50 sites in my *LastPass* vault. My security rating is less than 40%. My first step will be to raise the rating. I feel that many sites, like MacMall, don't really need a secure password; I keep no credit card information there. Nonetheless, each site needs a unique password so that the compromise of one password doesn't inadvertently compromise many more sites. It seems easier to just let *LastPass* generate passwords for me.

In Figure 2 you'll notice several tabs across the top. I have been using only VAULT to this point. I am intrigued to try SHARE next. Why? My activity

of recent days has virtually destroyed my wife's access to our shared sites. The SHARE option appears to give me a secure way to send some of the contents of my vault to her vault, albeit only one site at a time. I want to know more about the security of this transfer first though.

LastPass Support

LastPass has a 100+ page online User Manual that provides access to YouTube videos (see URL above), and staffs a HELP desk. The couple of times I emailed the HELP desk, the response was prompt and helpful. That's the plus. Both the manual and the videos, on the other hand, need work. When I went to the online manual for some explanation, I found the information provided wasn't detailed enough. I had a similar experience with the one or two videos I watched — too terse and poorly produced. Looking back I feel like I had to figure out most of the ins and outs of *LastPass* for myself.

Conclusion. I think *LastPass* is a workable solution to the problem of password management. It shouldn't take long to find out; in the coming weeks, I'm obliged to use it exclusively to access my newly protected sites.

Bob Whitesel has been a member of Washington Apple Pi since 1996.

He welcomes questions and comments on this article at:

bwhitesel39@wap.org. LastPass is a free download from www.lastpass.com.

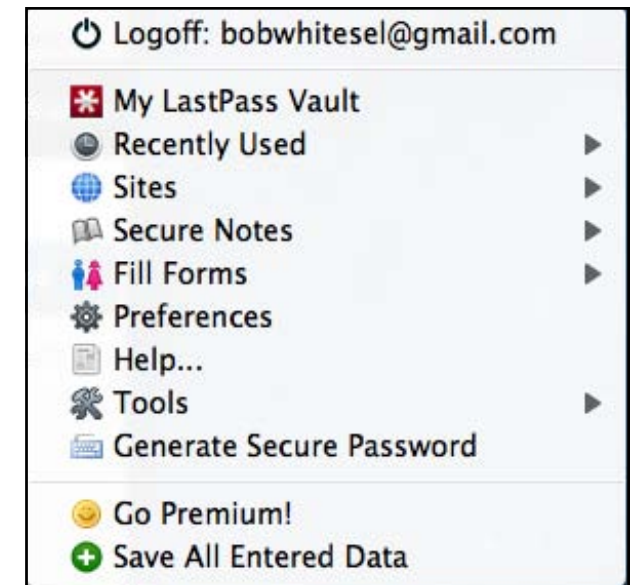


Figure 3: Main menu for the *LastPass* app.

FlippedBITS: IMAP Misconceptions

Joe Kissell

In today's installment of FlippedBITS, I want to examine a handful of common misconceptions about IMAP, a familiar protocol for retrieving email from a server. IMAP stands for... well, thereby hangs the first tale. IMAP's inventor, Mark Crispin (who, sadly, died in December 2012), called the first version of his creation Interim Mail Access Protocol. Versions 2, 3, and 2bis were referred to as Interactive Mail Access Protocol, and version 4 — what's in use today — is officially Internet Message Access Protocol. Although many Web sites claim that the acronym once stood for Internet Mail Access Protocol, I have found no credible references to back up that claim.

By whatever name, IMAP has always been a means by which email clients can talk to email servers. That puts it in the company of POP (Post Office Protocol) and Microsoft's MAPI (Message Application Programming Interface). Almost every modern email client — including Apple Mail, Thunderbird, Microsoft Outlook, and dozens of others — supports IMAP as a means of retrieving email. You may very well have been using it for years without even knowing it — iCloud and its predecessors MobileMe, .Mac, and iTools have always defaulted to IMAP for email access.

New Kid on the Block. The first thing I want to clear up is the persistent notion that IMAP is some sort of newfangled email system, a regular Johnny-come-lately compared to the ancient and revered POP method. Yes, POP has been around quite a while — it was invented in 1984. IMAP came along in 1986. (For perspective, Apple's Macintosh System Software 5 — the first one to include

MultiFinder — was released in 1988.) Both protocols subsequently underwent numerous revisions, but in any case, it's a bit silly to consider POP "traditional" and IMAP "new."

Now, it's true that in the early days, email clients and servers alike were more likely to support POP than IMAP (and even today, IMAP support isn't universal). So, many of us who have been using the Internet for a long time became accustomed to POP — and a surprising number of people still use POP, often out of habit more than necessity. (I'll return later to whether that's a good idea.) But IMAP has been a viable option for decades.

Are You Being Served? The usual way people explain the difference between POP and IMAP is to say that with POP, all messages are downloaded from the server to your email client, whereas with IMAP, messages are stored on the server. That's sort of true-ish, but it's unfairly misleading in both cases. With POP, you can leave messages on the server if you want to, and with IMAP, you can download all your messages and store them locally. The simplified "IMAP-means-stored-on-the-server" explanation has led countless people to assume that you can use IMAP only when you have an active Internet connection. But that isn't the case. For as long as I've been using IMAP, I've maintained local copies of every single message in my accounts, and have never had trouble reading, searching, filing, or otherwise managing my messages when offline.

The best way to think about IMAP is that the server holds the master copy of every

message. Whenever an IMAP client connects to the server, it can synchronize changes bidirectionally — for example, new messages in the Inbox download to the client; changes made in the client while it was offline upload to the server, updating the master records. But the exact behavior is determined by the design of the client and settings chosen by each user. By default, Apple Mail (like most other modern email clients) keeps all your messages in sync between client and server. But if you prefer, you can configure your client not to cache messages for offline viewing, to cache only some messages, or to cache the text of messages but not any attachments.

I should add that even though the server stores all your email messages, this in no way prevents you from deleting messages. Although, again, the exact behavior varies according to your client and your settings, when you delete a message locally, your client normally tells the server to delete its copy too.

I'll File Away. Another prominent difference between POP and IMAP is that IMAP lets you define mailboxes (that is, folders for email messages) that are stored on the server and (in most cases) synced with your email client. In general, the effect is that no matter which IMAP client you use, on which platform, it will always reflect the same set of mailboxes with the same contents; you'll never have to worry that you might have filed a certain message on the wrong computer.

With POP, there's no such thing as server-based mailboxes, just an Inbox, so any filing you do must, by definition, be done in the client. However, with IMAP, even though server-side mailboxes are supported (and quite handy), if you prefer to store some or all of your messages in local mailboxes, nothing's stopping you.

In fact, if your IMAP provider imposes a storage quota, you may want to move messages from server-based mailboxes into local mailboxes from time to time in order to free up space on the server.

The Same Thing, Only Different. I've heard it said that if you configure your POP client to keep all messages on the server — that is, not to delete them after they're downloaded — then POP becomes so similar to IMAP that you probably won't be able to tell the difference. But that's very far from the truth.

Apart from the lack of server-based mailboxes (which, of course, you're not obligated to use in IMAP), leaving messages on a POP server is much different from leaving messages on an IMAP server. Crucially, IMAP servers keep track of which messages you've read, replied to, and forwarded. So, suppose I connect to a POP account that has 15 messages in the Inbox. I download and read them, but leave them on the server. Now I go to a different client or computer and connect to the same POP account. The same 15 messages will download again (along with any that have arrived in the meantime), with no indication of which ones I've already read. By contrast, if I do the same thing with a pair of IMAP clients, each one will show me the same thing — these messages have been read, those haven't; this one has been replied to; that one was forwarded; and so on. This makes it much easier to switch among clients — something that becomes increasingly important as more of us have not only multiple computers but also smartphones, tablets, and other Internet-connected gadgets.

Speaking of multiple clients, you should be aware that POP permits only one connection at a time per account, while IMAP has no such limit. So, although your

three Macs, two iPads, Windows PC, and iPhone can all maintain live connections to an IMAP account, they're forced to take turns with a POP account.

All of a Piece. But now, let me turn that around and address another misconception, that all IMAP servers are created (more or less) equal. Would that it were so, but no. IMAP servers are as frustratingly different from each other as clients are. It all comes down to three words: specification, implementation, and configuration.

The IMAP specification, as I mentioned earlier, has undergone a number of revisions. In addition, it supports the use of optional extensions to provide extra features. When it comes time to implement the specification, one developer might use an older version of the spec, or interpret part of it in an idiosyncratic way, or choose to include or omit various extensions for one reason or another — while the next developer might make entirely different choices. And some developers might decide that the standard IMAP approach doesn't meet their needs, so they leave things out, slap extra things on, and rejigger other things so they work in surprising ways. (This happens more often than I'd like to admit, although Gmail's flavor of IMAP is arguably the least IMAP-like, which is not surprising since it was an afterthought rather than a part of the original Gmail design.) Moreover, IMAP servers have a variety of settings a system administrator can configure, just as IMAP clients have user-configurable preferences. All these variables can make any IMAP client/server pair behave much differently from any other.

I can't tell you how many times I've had to say things like, "Yes, your IMAP server

supports subscribing to specific mailboxes, and so does Outlook, but Apple Mail doesn't," or "Apple Mail supports IMAP IDLE (see "How Apple Mail May Be Anything but IDLE when Pushing Email [1]," 22 October 2012) but your IMAP server doesn't," or "Gmail's idea of archiving bears only the remotest resemblance to Apple's idea of archiving." One especially troublesome area is the way various IMAP servers and clients handle deleting messages — a messy topic I address somewhat in my books about Apple Mail ("Take Control of Apple Mail in Mountain Lion" and "Take Control of Mail on the iPad, iPhone, and iPod touch [2]") but won't delve into further here.

POP on over to IMAP. Notwithstanding the several quirks and annoyances of certain IMAP implementations, my fondness for IMAP is right up there with my fondness for chocolate. (That's way up there, in case you were wondering.) Let me summarize the advantages of IMAP over POP:

- The server keeps a master copy of all your data (including mailboxes and message metadata such as read or replied). So you'll see the same thing with any client on any platform.
- You can connect to an IMAP account from multiple clients at the same time.
- If your client supports it, you can have it download only message headers, with full message bodies on demand.
- You can ask your client to search for messages on the server, even if they haven't been downloaded. (The iOS version of Mail supports this, but the Mac version doesn't.)

The oft-heard objection to IMAP that it takes away one's control is a myth. As long as you have your client configured to cache a local copy of all messages and to

delete messages on the server when you delete them locally, you maintain just as much control over your email as you do with POP. Most of the old assumptions that led users to favor POP — such as the expectation that a person will use a single computer for email most of the time, and the belief that online storage is expensive — are no longer valid in today's world.

Are there still legitimate reasons to use POP? Sure. For one thing, it's less chatty than IMAP, so it tends to be better in low-bandwidth situations, especially when lots of users are connecting to an underpowered server. (Having said that, mobile IMAP clients typically manage to do a great job even over slow cellular connections, but that assumes optimization of both client and server for that purpose.) Also, most providers cap each user's IMAP storage quota, so if you have vast amounts of stored email, you may be forced to offload some of it to local mailboxes; by contrast, POP normally holds onto messages only until the user picks them up, so its storage requirements tend to be lower. And, if you're concerned that your email provider can't be trusted or is vulnerable to hacking, you might prefer not to keep unencrypted email on a server any longer than necessary. Finally, not all email providers support IMAP. But that leads me to my final point.

Stuck in the Past. I've heard from a number of people who tell me they'd like to use IMAP, but they can't, because their ISP doesn't support it — or charges extra for it. So, two things here.

First, even if your ISP doesn't offer IMAP for accounts on its own email server, that in no way prevents you from using another IMAP provider. When an ISP says they

charge extra for IMAP, that means they charge extra to use their IMAP server, not any IMAP server. You can go right ahead and use iCloud, Yahoo Mail, AOL, Gmail, or any of a hundred other services that offer IMAP access to email — many of which are free.

Second, if your main email address comes directly from your ISP, and that ISP doesn't support IMAP, you can usually set up an IMAP account with a different provider and then forward mail from your ISP to the new IMAP account. (Exact directions to do this depend on the provider.) That way, anyone with your old address can still reach you, while you get to enjoy the advantages of IMAP.

If you've weighed the pros and cons and decided that a switch from POP to IMAP is for you, see if your existing email provider offers an IMAP option — sometimes it's as simple as flipping a switch on the server side, although you'll likely have to configure an entirely new account in your email client. For further guidance, I recommend Kirk McElhearn's Macworld article "How to convert a POP email account to IMAP [3]."

[1]: <http://tidbits.com/article/13344>

[2]: <http://www.takecontrolbooks.com/iphone-mail?pt=TB1170>

[3]: <http://www.macworld.com/article/1165600/>

Permanent article URL: <http://tidbits.com/article/13692>. Unless otherwise noted, this article is copyright © 2013 TidBITS Publishing, Inc.. TidBITS is copyright © 2008 TidBITS Publishing Inc. Reuse governed by this Creative Commons License: <http://tidbits.com/terms/>.

What version of OS X should I run?

Jimmy Obomsawin



OS X 10.5 Leopard

Don't run this if at all possible. Apple no longer releases security fixes and most current software will not work.

The only reason to stay on Leopard:

- You are running a PowerPC Mac.

If you must continue using your PowerPC Mac I insist that you use a web browser such as TenFourFox to take advantage of recent security updates



OS X 10.6 Snow Leopard

Considered by traditionalists to be the best version of OS X. Very refined, but lacking some modern features such as iCloud support.

A few programs no longer support Snow Leopard.

Reasons for staying with Snow Leopard:

- You have a 32-bit Intel computer
- You don't care about iCloud syncing
- You run older PowerPC software such as Office 2004 or Adobe CS2



OS X 10.7 Lion

In my experience it's kinda buggy, slow, and though it supports iCloud the integration isn't as seamless as it is on Mountain Lion.

As far as I'm concerned there is only one reason to stick with Lion:

- Your computer doesn't support Mountain Lion but you want iCloud integration.



OS X 10.8 Mountain Lion

A refinement of Lion. Only easily apparent new feature is better app parity with iOS. In other words, just like on iOS you have separate Calendar and Reminders apps. But the whole OS is sprinkled with little tweaks for better usability and reliability.

Most Apple users have multiple Apple devices so I usually recommend running this version for an integrated experience across all devices. There have been three major updates for Mountain Lion already, so most of the major bugs have been kicked.

FlippedBITS: Java, JavaScript, and You

Joe Kissell

Lately there have been a lot of news reports about security concerns with Java, Java-related software updates from Apple, and numerous conjunctions of the words “Java” and “danger.” At the same time, I’ve observed an enormous amount of confusion over what Java is, what the potential problems are, and what the consequences of getting rid of Java might be. There’s further confusion over Java updates coming from Apple versus Java updates coming from Oracle; what a Java Runtime is; how JavaScript relates to Java (spoiler: not at all), and more. In this installment of FlippedBITS, I’m going to attempt the quixotic undertaking of sorting all that out for you.

Let’s start at the beginning.

Java, East of Krakatoa — Java is the name of the fifth-largest (and most populous) island in Indonesia. I’ve been there a couple of times, most recently when I turned 40. My wife and I hiked up to the rim of Mt. Bromo, an active volcano, at sunrise on my birthday. Come on over some evening and we’ll show you our slides over a nice cup of... java. It so happens that a great deal of high-quality coffee is grown on the island of Java, hence the nickname. (It also so happens that I single-handedly consume 3.5 percent of the world’s coffee; hence another nickname for coffee, “Joe.”) In the early 1990s when a team of engineers at Sun Microsystems was developing a new programming language, they toyed around with several names before settling on Java, allegedly because they, too, were coffee enthusiasts.





Let's just say that as programming languages go, Java is a pretty nice one. It's powerful, popular, and — crucially — designed in such a way that once a Java application is compiled, it can run on many different platforms.

So, for our purposes, Java is a programming language. I could tell you that it's an object-oriented language largely based on C++, but if you're a programmer you already know that, and if you aren't, you wouldn't care. Let's just say that as programming languages go, Java is a pretty nice one. It's powerful, popular, and — crucially — designed in such a way that once a Java application is compiled, it can run on many different platforms. That's right, a given Java application can run on a Mac, a Windows PC, a Linux PC, or a smartphone without any modifications. (In practice, that's a bit of an oversimplification, but it's a convenient fiction.)

How does Java pull off this feat of legerdemain? It relies on something called a virtual machine. If you've ever run Windows or Linux on your Mac using an application like Parallels Desktop or VMware Fusion, you already have a general idea of what a virtual machine is — it's an environment, created in software, that functions like a physical computer. Just as a Windows virtual machine lets you run Windows on a Mac (or even within another copy of Windows), the Java Virtual Machine (JVM) lets Java software run on any platform. Each host platform has a different JVM that's designed to run on its physical hardware — for example, Intel x86 chips have one JVM, while ARM chips have a different one.

Now, there's a little more to it than that, so please bear with me

for two slightly technical paragraphs.

First, when I say the JVM lets "Java software" run on any platform, the software I'm referring to is what's known as Java bytecode. Java bytecode isn't Java as such, but rather a sort of intermediate language created when Java code is run through a program called a compiler. Ordinarily, this distinction wouldn't be important to a non-programmer, except it turns out that other programming languages besides Java can also be compiled into Java bytecode, and then run by the JVM. So, someone could write a program in, say, Python or Ruby, and use a special compiler to build that into something that, as far as the JVM is concerned, is indistinguishable from a program written in Java. In this article, we're concerned with any software that runs in a JVM, regardless of what language it was written in.

Second, a JVM by itself is usually not enough to enable Java bytecode to run on a computer. You also need a platform-specific version of the Java Class Library, which tells the JVM how to do particular tasks on that platform. For example, maybe a Java program contains an instruction to play the system beep sound. Mac OS X does that one way, while Windows does it another way. So, the Java Class Library takes an instruction that the JVM is trying to send to the host platform and passes it on in the form the host platform expects. The JVM and the Java Class Library are almost always distributed

together as a package, and that package is known as the Java Runtime Environment (or JRE), commonly shortened to “Java Runtime.” The Java Runtime is sandboxed (much like iOS and Mac App Store apps), which was supposed to help with security, but secure sandboxes are extremely difficult to develop, and the Java sandbox hasn’t fared well — I’ll return to that issue shortly.

To sum up thus far: Any device with a Java Runtime installed can run Java bytecode, which may have been originally written in Java or some other language.

Once Upon a Platform — In the early days of Mac OS X, Apple not only included a built-in Java Runtime (licensed from its then-owner Sun), it actively promoted Java as a “first-class citizen.” Developers were free to write their applications in Objective-C, Apple’s own programming language that was originally part of NeXTSTEP, or in Java. Either way, users would end up with a valid application that looked and felt (more or less) native. (Java apps have often been criticized as feeling “not quite right” because they often use interface elements that are different from those of native Mac apps, but that’s a relatively minor point.) As a result, lots of Mac apps were — and a few still are — written in Java.

Java isn’t just for stand-alone, double-clickable applications, mind you. A Java applet can also be embedded in a Web page. Assuming your computer has a Java Runtime installed, your browser has a Java plug-in (to support embedded applets), and Java support is enabled, highly complex programs called applets can run right inside your Web browser. Before Flash and Silverlight began to catch on, Java applets were a common way to add interactivity and complex computational capabilities to Web pages.

But over the years, Java has gone from first-class citizen to suspiciously regarded foreigner (and not just on the Mac). The whole story is long and twisted, involving a combination of technical, legal, and political issues. I’ll hit just a few recent highlights.

Java — including the tools to develop and compile it, the runtime environments, and various other pieces — has been open-source since at least 2007, but it’s maintained primarily by Oracle Corporation, which acquired Sun Microsystems in 2010. Although Oracle’s implementation of Java isn’t the only one, it’s as close as you can get to the “official” version. For a long time, the version of the JRE Apple included with Mac OS X was always several months or more behind Oracle’s latest version. And this was a problem when, for example, a security flaw was discovered. Oracle might fix it quickly, but Macs remained vulnerable for some time, until Apple caught up.

Let’s talk about those security flaws for a moment. I’m sorry to say the Java Runtime has had a lot of serious security problems, and more turn up all the time. (To be precise, Apple’s Java updates in 2013 alone address 56 unique vulnerabilities.) Notice that I said Java Runtime — it’s not the Java programming language itself that has issues, but rather the environment used to run Java bytecode. Even then, the real problem isn’t the Java Runtime as such, but rather the fact that if your Web browser has a Java plug-in installed and enabled, and you happen to visit a Web page that contains a malicious Java applet, it can do all sorts of serious damage. Some of the flaws enable Java code that’s supposed to stay safely within your Web browser to jump outside the sandbox, as it were, and cause all sorts of mischief elsewhere on your computer. It’s nasty, nasty stuff.

And the bad guys have been working overtime to find and exploit these security holes.

Java for OS X 2013-003 and Mac OS X v10.6 Update 15

Java

Available for: Mac OS X v10.6.8, Mac OS X Server v10.6.8, OS X Lion v10.7 or later, OS X Lion Server v10.7 or later, OS X Mountain Lion v10.8 or later

Impact: Multiple vulnerabilities in Java 1.6.0_43

Description: Multiple vulnerabilities existed in Java 1.6.0_43, the most serious of which may allow an untrusted Java applet to execute arbitrary code outside the Java sandbox. Visiting a web page containing a maliciously crafted untrusted Java applet may lead to arbitrary code execution with the privileges of the current user. These issues were addressed by updating to Java version 1.6.0_45. Further information is available via the Java website at <http://www.oracle.com/technetwork/java/javase/releasenotes-136954.html>

Apple has used multiple tactics to address these problems, and for some time now has been trying hard to push users in the direction of not using Java at all.

Starting with Mac OS X 10.7 Lion, Apple no longer includes a Java Runtime with the operating system, but if you try to run a Java app, your Mac prompts you to download and install Java Runtime – it’s a matter of a few clicks. What you get if you do that is not the latest release. Apple gives you a version of Java 6 (that is, build 1.6.x), whereas the latest from Oracle is Java 7 (that is, build 1.7.x). If you want Oracle’s version, you can [download it](#) [4], and installing it will override Apple’s version. But you probably shouldn’t do that, because Java 7 has had even more security issues than Java 6. For the time being, Apple is actively updating its version of Java 6 with security patches, while Oracle is maintaining Java 7 with comparable fixes. And, unlike in past years, Apple is now delivering many of those patches just as fast as Oracle. In addition, Apple has blocked Safari from using certain particularly vulnerable versions of the Java plug-in. (Meanwhile, Java isn’t available at all on iOS, and you can see why.)

Joe on Java — I want to reiterate two main points to be sure they’re crystal clear. On the one hand, neither the Java programming language nor the Java Runtime will hurt you or your Mac. Merely having the Java Runtime installed does not introduce any security risks. In fact, even running stand-alone Java applications is safe, as long as they come from well-known sources. Or, to put it differently, it’s just as safe to run a stand-alone Java app as it is to run any other app (because, after all, any app could in theory be compromised).

On the other hand, having Java enabled in your browser, at this point, is wildly dangerous. I strongly suggest turning it off. To do this in Safari, choose Safari > Preferences, click Security, and uncheck Allow Java. In Chrome, visit <chrome://plugins> and click the Disable link underneath Java. In Firefox, choose Tools > Add-ons, click Plugins, and click the Disable button next to the Java Applet Plug-in.

If you’re using the latest version of Safari, you can enable Java selectively for individual Web sites (leave Java enabled, but then agree to each site’s usage of Java individually if you’re sure it’s safe; for details, read “[Safari Updates Add Extra Layer of Java Protection](#) [5],” 26 April 2013). But the number of Web sites that legitimately use Java these days is small indeed, and I suggest leaving Java off in your browser unless you’re absolutely certain you need it.

Now, in case you’re wondering if you should go ahead and uninstall the Java Runtime altogether, I’ll lay it out for you. If you’re running Lion or later, you’ll have the Java Runtime on your Mac only if you tried to run a Java app (in which case, if you still want to run that app, you still need Java Runtime) or you downloaded it from Oracle yourself (again, presumably because you needed it). If you’re

running CrashPlan, which I strongly endorse, you currently need Java. (CrashPlan developer Code 42 Software is working on a non-Java version of CrashPlan for Mac, to be released later this year.) Portions of Adobe Creative Suite, including Photoshop, rely on Java. So do OpenOffice, a few games, and a handful of productivity apps. If you need an app that relies on Java, you must hang onto the Java Runtime. Stick with Apple's version of Java, and turn it off in all your Web browsers.

If you don't need Java but still have it installed, you can uninstall it. Rich Mogull has instructions for either uninstalling or disabling it, as the situation warrants, in his Macworld article [“How to disable Java on your Mac \[6\].”](#)

JavaScript — So that's Java. But now we come to another coffee-like computing term: JavaScript. JavaScript is the name of another programming language, originally developed by Netscape. It bears only a vague resemblance to Java, in that both languages drew inspiration from the much older language C. The similarity in names was essentially a marketing stunt. JavaScript was previously called LiveScript, but Netscape apparently wanted to capitalize on the recent popularity of Java, so the company renamed it JavaScript. That's a shame, because from day one people have assumed that JavaScript was somehow related to Java, but it isn't. It's just another language.

Unlike Java, JavaScript doesn't rely on a virtual machine. However, it is an interpreted language, which means it doesn't create stand-alone applications, or even applets. Software called an interpreter has to read the raw programming code and execute its instructions on the fly. JavaScript is most often used to add features to Web pages, so virtually every Web browser includes its own JavaScript interpreter.

JavaScript can do an amazing number of things on a Web page, including many of the tasks Java was previously used for. Lots of sites have dynamic menus and other navigational controls created with JavaScript. Photo galleries, Web apps for email and calendars, word processors like Google Docs, and many other common tools rely heavily on JavaScript. You can disable it in your browser if you want to. But don't. It's such a useful and pervasive tool that your experience of the Web will become quite poor without it — many sites may not even be usable at all. (And, I might add, JavaScript is available even on iOS devices.)

That's not to say JavaScript has a perfect security record. Certainly it can be used for lots of annoying things, such as pop-up ads and resizing windows. But JavaScript's threat level can't compare to that of Java, because JavaScript can't reach outside your browser.



That's not to say JavaScript has a perfect security record. Certainly it can be used for lots of annoying things, such as pop-up ads and resizing windows.

Final Thoughts — I'm quite fond of Java (the island, the drink, and even the programming language). But I've turned off Java in all my browsers, and when the Java software I depend on has been replaced with native Mac versions, I'll uninstall Java and never look back. Java's "write once, run everywhere" approach is brilliant in theory, but in practice isn't worth the hassles. As for JavaScript, it's all good — but don't be surprised if I tweak my browsers to block pop-up windows and other annoying behaviors.

[Java map by Burmesedays. [CC-BY-SA-3.0](https://creativecommons.org/licenses/by-sa/3.0/) [7], via Wikimedia Commons.]

[1]: http://tidbits.com/resources/2013-05/Java_region_map.png

[2]: http://tidbits.com/resources/2013-05/Java_logo.png

[3]: http://tidbits.com/resources/2013-05/Java_multiple_vulnerabilities.png

[4]: http://www.java.com/en/download/mac_download.jsp

[5]: <http://tidbits.com/article/13716>

[6]: <http://www.macworld.com/article/2028900/>

[7]: <http://creativecommons.org/licenses/by-sa/3.0>

Permanent article URL: <http://tidbits.com/article/13730>

Unless otherwise noted, this article is copyright © 2013 TidBITS Publishing, Inc.

TidBITS is copyright © 2008 TidBITS Publishing Inc.

Reuse governed by this Creative Commons License: <http://tidbits.com/terms/>.



The New Improved Pi Calendar

Want to know when and where you can find the next Washington Apple Pi meeting? Well look no farther than the recently renovated Pi Calendar. Go to the home page (www.wap.org) and click on Calendar in the toolbar. The calendar will reveal the scheduled Pi activities for three months; this month and the two following months. Hover your cursor over the event and you will see a description of the event as well as the location and time. Better yet, scroll down to just beneath the present month calendar and you will see this link:

Tip: [Subscribe with your calendar software to get automatic updates](#)

Washington Apple Pi Meetings



Rob Pegoraro will be our speaker for the **July 27th** General Meeting at **George Mason University**.

From his website:

Rob Pegoraro tries to make sense of computers, consumer electronics, telecom services, the Internet, software and other things that beep or blink through reporting, reviewing and analysis—from 1999 to 2011 as the Washington Post’s tech columnist, now for a variety of online and print outlets.

As always, check the [Pi website](#) for additional, updated details on these meetings.

Members Helping Members

Dedicated volunteers are the lifeblood of the Pi. Whether it is installing a hard drive at a Clubhouse Saturday event, answering questions at a General Meeting, or placing advice on the TCS Forums, there are many ways to pitch in and help, depending on a member's interests and talents. Please consider making a commitment to ensure that the Pi remains a strong and energetic club dedicated to helping its members. All volunteer efforts are appreciated. Check out these areas where assistance is needed.



Featured Position – General Meeting Webcast Director

The Pi currently captures its General Meetings on videotape for future viewing by members who cannot attend the meetings. Using the same camera, we are able to concurrently produce live webcasts so members can view the meeting in real time via the Internet. We need a dedicated volunteer to oversee the technical and production aspects of this webcast.

There are various technical components to putting up the video on the Pi's Ustream web page. The Webcast Director needs to ensure that the equipment is properly set up and functioning, and needs to monitor the audio and video streaming production from start to finish (or designate someone to do this) to ensure satisfactory audio and video quality. The Webcast Director will not be responsible for any other part of the General Meeting program, but will focus exclusively on the webcast production.

Any previous experience in TV or video production would be advantageous, but is not required. A basic understanding of video and audio equipment and how to connect them would be helpful, but the contracted cameraman has primary responsibility for equipment hookup. The Webcast Director needs to make sure that the meeting presenters do not stray from their predetermined stage positions, so they remain adequately illuminated for the entire presentation. Also, this person should be available in case of a technical or other problem that would require the speaker to temporarily suspend their presentation. Case in point: the Webcast Director will stop the action when the cameraman indicates that a new tape needs to be swapped out in the video camera.

If you are interested in helping out with an eye on taking over this important position, contact office@wap.org.

Journal Managing Editor

The *Journal* is an established, time-honored publication of the Washington Apple Pi and it is in need of someone to take over as Managing Editor, leading a team of dedicated and highly skilled volunteers. The *Journal* is published on a bi-monthly basis so most of the activity occurs in the three weeks leading up to the publication date. In addition, this person will be a member of the Pi's Publications Committee, which has oversight of all printed material published by the Pi. The job is challenging, but very rewarding.

If you enjoy writing and want to help the Pi progress, this is the job for you. If you enjoy working with enthusiastic colleagues on meaningful projects, apply today.

Contact the interim *Journal* editor, Jay Castillo, at: editor@wap.org with any questions or to express your interest.



General Meeting Manager

Our General Meeting programs need someone dedicated to the planning and managing of these important club events. The job includes coordinating meeting activities, making any necessary arrangements, and managing the behind-the-scenes tasks. While it sounds like a job for Superman, that really is not the case. If you have juggled household duties or been an office administrator, you have the skills. While another Pi manager emceeds the meeting, you will be making him or her look good!

Because this activity needs to be approved by the Pi's Management Committee, the volunteer will serve as an adjunct member of that committee. This volunteer opportunity is guaranteed to keep you busy, and when the meeting is over and everything is packed up, lunch is on the Pi.

Send your inquiries to office@wap.org to express interest and we'll go from there.

iChat Help Volunteer

Apple's *iChat* application offers a way for Pi members to help other members from the comfort of their own home. Interested? Read on.

iChat is an ideal means for direct communication using text, voice, video, or, in the last two versions of Mac OS X, screen sharing. With the Pi having commercial grade servers and server software, we now have the means to host our own *iChat* network. We are currently testing the capability of this interactive networking software, and the first results are positive.

So what will Pi *iChat* Help volunteers do? Basically, the volunteer helps members who have questions or a problem, but instead of meeting face-to-face, you handle the matter using *iChat*. What's that? You don't use *iChat* or haven't even launched it yet? No problem, we will train you. We will also be providing guidance on how to deal with common issues, and on problem-solving techniques, and will supply a list of associate helpers who can help with particular issues. All you need to bring to the equation is your people skills and a genuine desire to help.

If you have even the slightest interest or any questions, contact office@wap.org.

Washington Apple Pi

