ComputorEdge
Online

Signing Digital Documents

January 23, 2015

List of *ComputorEdge* Sponsors

San Diego *ComputorEdge* Sponsors

Colocation and Data Center

redIT

With approaches like smart security, customized colocation and an extensive range of managed services, redIT helps you intelligently leverage IT.

Macintosh Specialists

Maximizers

Serving San Diego County Since 1988 * Onsite Macintosh Service for Home and Small Office Needs * ACSP: Apple Certified Support Professional ACTC: Apple Certified Technical Coordinator Apple Consultant's Network

Repair General

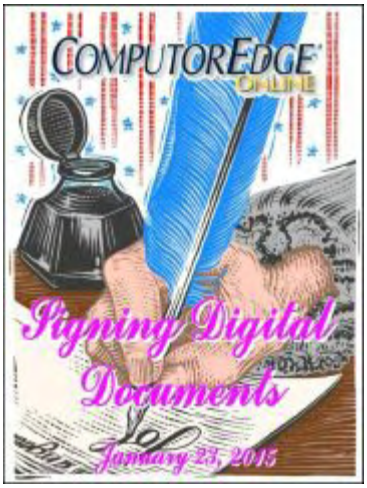
Hi-Tech Computers

Notebooks, Monitors, Computers and Printers We Buy Memory, CPU Chips, Monitors and Hard Drives Windows 7 Upgrades Phone (858) 560-8547

Colorado *ComputorEdge* Sponsors

ComputorEdge™ Online — 01/23/15

Click to Visit *ComputorEdge™* Online on the Web!



Signing Digital Documents

Scanners help eliminate paper, but officially signing documents is still a little tricky.

Magazine Summary

List of *ComputorEdge* Sponsors

Digital Dave
by Digital Dave
Digital Dave answers your tech questions.
Password Management Programs?; How to Keep a Safe Password List.

Scanning Makes Us Paperless—Almost
by Jack Dunning
Scanners Are Now as Important as the Printer
While combo printer/scanners are commonplace, businesses continue to wrestle with the best way to digitally sign important documents.

Understanding the Mysteries of Regular Expressions (RegEx) in AutoHotkey
by Jack Dunning
To Understand How a RegEx Works, It Helps to See Trains Running Down a Track
Many AutoHotkey script writers don't use Regular Expressions because they seem too mysterious and confusing. All they really need is a little understanding.

Wally Wang's Apple Farm

by Wally Wang

Interesting Uses for Mobile Computing

Interesting Uses for Mobile Computing; Swift: The Future Programming Language of OS X and iOS; February's Apple Event; The Decline of Google Glass; Another Casualty of the Print World; Apple's Cocoa and Cocoa Touch Framework; The Services Menu.

Editor's Letters: Tips and Thoughts from Readers

by ComputorEdge Staff

Computer and Internet tips, plus comments on the articles and columns.

"Backing up Files with Long Names/Paths," "Internet over Power Lines," "Scan with HP Officejet 8600," "Micro Power Now! (Circa 1983) Comments"

ComputorEdge Magazine, P.O. Box 83086, San Diego, CA 92138. (858) 484-1998

Click to Visit *ComputorEdge*™ Online on the Web!

Digital Dave

"Digital Dave answers your tech questions." by *Digital Dave*

Password Management Programs?; How to Keep a Safe Password List.

Password Management Programs?

Dear Digital Dave,

What's with Dashlane and LastPass? What if your hard drive crashes or your laptop is stolen and you have to buy a new computer? What if you sometimes log on from a public computer? It sounds very risky to me.

Here's an article about password managers.

Thanks. Love ComputorEdge *and always read your column first every Friday.*

Steve Johnson
Fort Lauderdale

Dear Steve,

If you use an online password manager, you are protected against the problems that occur with a hard drive crash or computer replacement. Since all of your passwords are stored online you can access them anytime from anywhere. As I understand it, when away from home or on another computer you can even access your sites via your password account, thereby not leaving any trail. You will need to remember the master username and password for your password account.

Security is number one on the list for these companies. Any breakdown, although always possible, would be catastrophic for their business. But even if they did get hacked, it is unlikely that your passwords would be compromised since, as I understand it, all of them are encrypted on your computer, not in the Cloud. Even the people at the password service would be unable to capture your data.

The one downside would be if the service went down, say, due to a denial of service attack. If you don't have an alternative backup, you may be out of luck for that period of time. For that

reason, I've included here as a tip, a manual system for creating and maintaining passwords which both gives an easy reference list, as well as protecting it from prying eyes.

It appears that the free versions of the programs only work through Web browsers. To save passwords for local desktop programs requires an upgrade.

Digital Dave

How to Keep a Safe Password List

While there are a number of online password sites which help you store your passwords, for whatever reason you may prefer to maintain your own list. If this is what you prefer, then one of the worst things you can do is make a list of programs/Web sites and passwords with Notepad or a word processor saving it on your computer. If you get hacked, then you are totally compromised.

Maybe you never save any passwords on the computer, but keep a handwritten list. This is totally safe from hacking, but if someone managed to get their hands on the list, you would have a problem. There is a manual approach that you can take to saving passwords which will both make it easy to remember the passwords and keep them safe—whether on the computer in an unencrypted text file or a hand written list. This will give you total control of your passwords in a secure way without dependency on a third party.

Dream Up a Master Root

The first step is to create a root which you will never write down or divulge to anyone else in the world—except maybe a very close family member just in case you get hit by a car. Memorize this root word. It might be something such as:

```
Dr3amR00t
```

which includes capital letters, a three, and a couple of zeros. This makes the root word more difficult to decipher while easily memorized. Remember, you are never going to write this down anywhere! It will only exist in your head—which is generally considered protected against the usual hacking techniques. (I suppose you could write it down and put it in a safety deposit box or a home safe. The word by itself on a piece of paper when discovered among your personal affects would be meaningless to most people.)

Every password you create will include this root word. This is the secret portion of your password and why it is so important that it never be written down. If someone knows this root then getting all of your passwords becomes trivial.

Create a Password Extension for Each Site and Program

This is the part of the password which will vary for each account or program which requires a password. You can write these down or save them to a text file. Otherwise, you'll never remember all of them. Whenever you create a new password you can either write down just the extension or add it to a dummy root. For example:

```
Bank       xxxxxxbank10
Costco     xxxxxxcostco15
Amazon     xxxxxxamz8
```

When logging in wherever the *xxxxxx* appears you will enter the master root (e.g. *Dr3amR00t*) —followed by the extension. (Just for a change of pace for some accounts you can also place the extension in front of the root, *n35xxxxxx*.)

Since you can write all of these extensions down, there is no need to memorize them, but if you use them a lot and have a mnemonic extension then you will find that you'll have little trouble remembering your most important (and most used) passwords. But even if you forget a password, it's simple to look them up on a piece of paper or in a text file. It is highly improbable that your passwords will ever be compromised—unless you divulge the root.

Trivial Web Site Accounts

There may be times when you will sign up for a Web site account where you have divulged no personal information (other than possibly an e-mail address) in order to get regular access. If someone hacked that account, you wouldn't care. Don't use your root password in any form, but rather use an easy to remember word that will give you access later—maybe your mother's maiden name.

Use this alternative easy password on all your trivial accounts. That way you won't ever need to write down or have trouble remembering your account password. Ask yourself, "Would I care if someone broke into this account?" If not, then use the simple password. If the answer is, "Yes!", then create a new extension for your memorized root and write down the extension. After a while you'll find that whenever you access one of these rarely visited sites you will always try the easy password first.

This manual system will work effectively without the need for any third party service. You could even maintain the list on Dropbox with very little risk. But its security depends upon keeping the master root absolutely secret. Don't be like the people who in recent on-the-street interviews gave it all away.

"What do you use for your password?"

"My grandmother's name."

"What is your grandmother's name?"

"Matilda."

More Safe Password Tips

You can also just use full sentences for passwords. The longer the better. They tend to be easy to remember and length is the most important factor for protecting against a brute force hack, not complexity.

For example, I could use:

```
thisisdavesfacebookpassword
```

as my Facebook password. Due to its length, it's improbable that it will be hacked, but it's easy for me to remember. I could use:

```
thisisdavestwitterpassword
```

for my Twitter account—if I had one—long, but easy to remember for multiple accounts. Naturally, if I used this type of reoccurring pattern, I would not disclose it to anyone I didn't trust.

If you really want account security, multi-factor authentication is the most important thing you can do these days. It minimizes the importance of keeping your password secret or complex.

This approach to account protection uses your e-mail account or, better yet, your cell phone to send you a code to relay if you log in to an account from a new location. This way, even if everybody knows your username and password, nobody can get into the account unless they either log in from your computer, can intercept a text to your cell phone number, or already have access to your e-mail account which, hopefully, is also secured with multi-factor authentication to your cell phone. The cell phone becomes like a digital ID card since only you should have your phone.

Scanning Makes Us Paperless--Almost

"Scanners Are Now as Important as the Printer" by Jack Dunning

While combo printer/scanners are commonplace, businesses continue to wrestle with the best way to digitally sign important documents.

There was a time when a digital scanner was a niche product for business and artists. But, now, almost anyone who owns a printer also owns a scanner. With today's printers, a scanner is part of the package. While most people buy three-in-one printers (printing, scanning, and copying) for printing, the scanner is rapidly becoming more important.

Not Paperless, But Less Paper

While our society is still a long way from paperless, we are certainly using less paper. Just look at all the trees that are saved by newspapers either cutting back on their runs or closing up shop. (That trend will continue. We've started calling the local newspaper from the major metropolitan area near where I live a pamphlet.) While I've cut way back on my personal printing, I still need a printer (with a scanner).

I like to do the *New York Times* crossword puzzle which runs in the local newspaper. Other than glancing at the front page and sports section, that puzzle is the primary reason I pick up the paper. (My spouse reads the whole thing.) When

delivery was cut back to four days a week, on the off days I could peruse the issue on my old iPad. But filling in the crossword on the iPad with a pencil didn't work at all. I started printing the puzzle on those paperless days. That was one piece of paper I previously had not needed to print. (It's a little ironic that by cutting back on paper use, the newspaper has forced me to use more paper—although on balance they save way more pulp than I use.)

Another common use of the printer is reproducing recipes. On the Web, you can find the steps for cooking up virtually any type of food. I know that it's possible to call up a menu masterpiece on a smartphone or tablet and directly use the device in the kitchen; however, I find it much simpler to merely print a copy of the recipe. Regardless of what anyone says, reading instructions from a handheld device is not ideal. Plus, the tablet or smartphone keeps turning off to save battery—especially when your hands are covered with sticky/mushy ingredients preventing you from turning it back on with anything other than your elbow. (Sorry, but even your iPhone fingerprint ID won't work in this scenario.)

The advantage of the paper recipe is that you can pick it up with messy hands without worrying about the resulting damage. But, even more important, is the fact that a printed copy won't go blank leaving you wondering what you should do next. Yes, when used selectively, there is definitely an important place for paper in our lives.

Enter the Scanner

The scanner turns the printer into a copier. Any documents can be fed into the machine for either copying or saving as a digital file. While making a quick paper copy is great, it's the scanner's ability to reduce paper to a file which is most important. Now you can actually eliminate the need to keep all of those paper copies (unless there is a need to maintain the original). Since most scanners will copy directly to PDF files, filing cabinets of bulky paper documents can be reduced to electronic files fitting on a flash drive or DVD disc. This is definitely a step in the right direction.

While in many ways PDF files are awkward to work with, they are probably the best file format to use. For one, PDF has become a universally accepted standard for transferring files via e-mail or download. With a scanner, any document can be quickly delivered directly to an addressee. Then, if necessary, the document can be printed on paper in the exact format as the scanned original. Otherwise, the PDF can be read directly on any computer or reading device. This is not the case with faxing.

Many of the current combo printers are capable of faxing (just plug in a phone line), but it is a less than ideal way to transfer documents. First of all, faxes generally go to a central location, then needing sorting and distributing. One of the most common complaints about faxes is that they get lost. "Never saw it. Could you fax it again?" Plus, if you inadvertently put the original in upside down, you will send blank pages. This is not an ideal solution.

Some people feel that sending signed copies via fax is somehow more secure than using PDF files. While faxes are maybe less likely to be intercept than a digital file, anyone who does gets their hands on the fax can easily scan it. However, faxing is a solution when documents need to be signed, but rather than reducing paper, they tend to create even more as the faxing cycle continues. My preference is for eliminating faxing completely—if possible.

Print, Sign, and Scan

When signing documents, one way to minimize the amount of print paper is to only print the signature page. Then, after signing, that page can be scanned and reinserted in the document. The problem is that many of the PDF reading programs do not facilitate swapping out the original page with the signed page.

I happen to own an old version of Adobe Acrobat which makes it easy to add, delete, or substitute pages. It is no problem for me to print, sign, scan, and substitute the signature page. As far as I can tell the free version of Foxit Reader does not have this capability (swapping pages)—although I'm sure that their paid PDF editing products do.

PDFill Free PDF Tools is a set of PDF tools which should do the job and gets pretty good reviews. (I've never used it myself.) According to those reviews, PDFill can be a little confusing and may take a while to master.

Here is a list of other free PDF tools which may facilitate page swapping. (PDFill is at the top of the list.) Look for merge and split features which will allow for the insertion (or replacement) of a new page.

The most important reason to print many documents is for adding signatures, then faxing it back or scanning and e-mail or, in some cases, uploading the document. If we can eliminate the need to print and sign, then we are one step closer to going paperless.

Electronic Signatures

Let's face it! There is not a good system for electronically signing documents. Even with all the protections built into any of the electronic signing methods, it is still possible for someone to scam the system. When working online, everyone is taking on faith that they are working with whoever the other person claims to be. An electronic signature doesn't make it any more valid. It's easy for someone to pretend to be someone else. That's one reason why there is such a problem with identity theft. The only way you have a chance of verifying a person's identity and true signature is to meet them in person and check their ID. Even then…

Having said that, you still want the capability to add your signature to electronic documents without first printing, then scanning again. Why print, sign, and scan, when electronically

adding your signature works just as well?

One of the best known companies which facilitates electronic signatures is DocuSign (watch the short video for how it works). For a price Docusign hosts documents sent by their clients for third party signatures. However, even though it is legally binding, it is not a person's real signature—although it may look similar. It is certainly convenient for everyone involved, but doesn't appear to be any more valid than clicking a button which says "I electronically sign." (I've done that a number of times for various reasons.) I'm not sure that DocuSign is any better than print, sign, and scan or the next technique—add your own signature directly to a PDF with PDF Sign.

Using PDF Sign to Save Paper

PDF Sign is built into Foxit Reader (as similar features are probably found in other PDF reading programs). It allows you to digitized your signature and apply it directly to a PDF document. In most cases this will be sufficient for an agreement, although it would be easy for anyone to sign your name if they can get a hold of your computer.

Before you can add a signature, you must create one. First, select "Create Signature" from PDF Sign (see Figure 1).

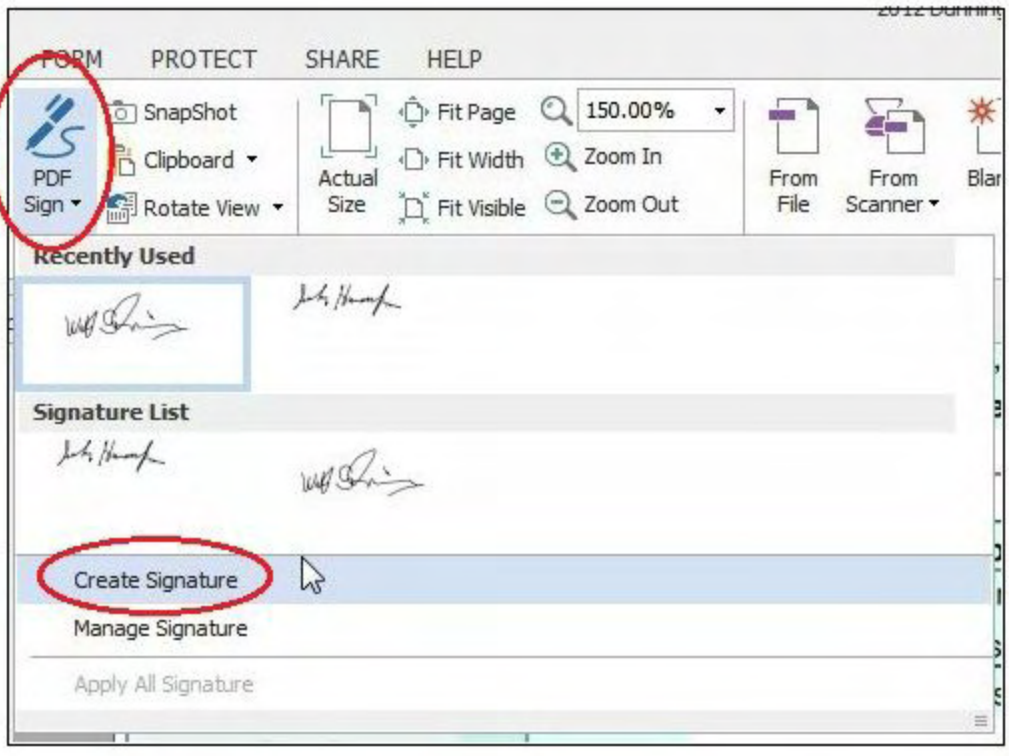


Figure 1. Click PDF Sign to use or create signatures.

Next, select Draw Signature from the dropdown menu and click Draw to write your name (see

Figure 2). If you have stored your signature to the Windows Clipboard (copied from somewhere else) or have a saved image file, then use the appropriate option.

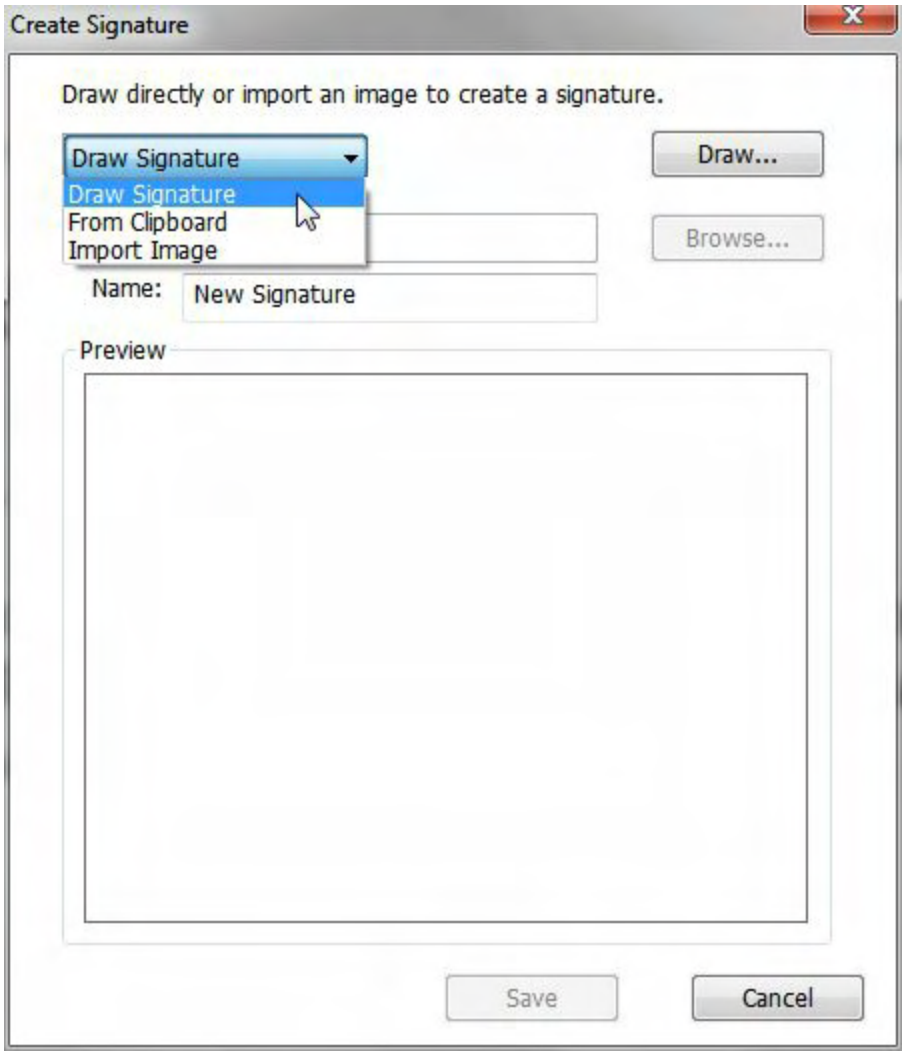


Figure 2. Select Draw Signature from the dropdown menu and click Draw to write your name.

The Draw Signature window opens (see Figure 3). Here you can use the mouse or other pointing device to write your signature. I found it particularly difficult to write with a mouse. I have enough problems with a pen or pencil. The example signature of John Hancock looks nothing like my handwriting.

Figure 3. Sign your name in the space provided with the mouse or other pointing device.

If I were serious about applying my signature to PDF documents, then I would most likely sign a piece of paper and scan it to an image file. Then I would have something that actually looks like my John Hancock. I might even scan a number of versions of my signature for those documents which require multiple signatures. That way the signature variations would make it look more authentic—even if it has exactly the same validity.

The last step is to apply the signature to a document. Pick the one you want from the list shown in Figure 1, then click on the appropriate place on the document screen. The signature will pop-up with the words "Double click to apply signature, right click for more options" (see Figure 4). You will be able to dragged the signature to the proper location and resize it as necessary.

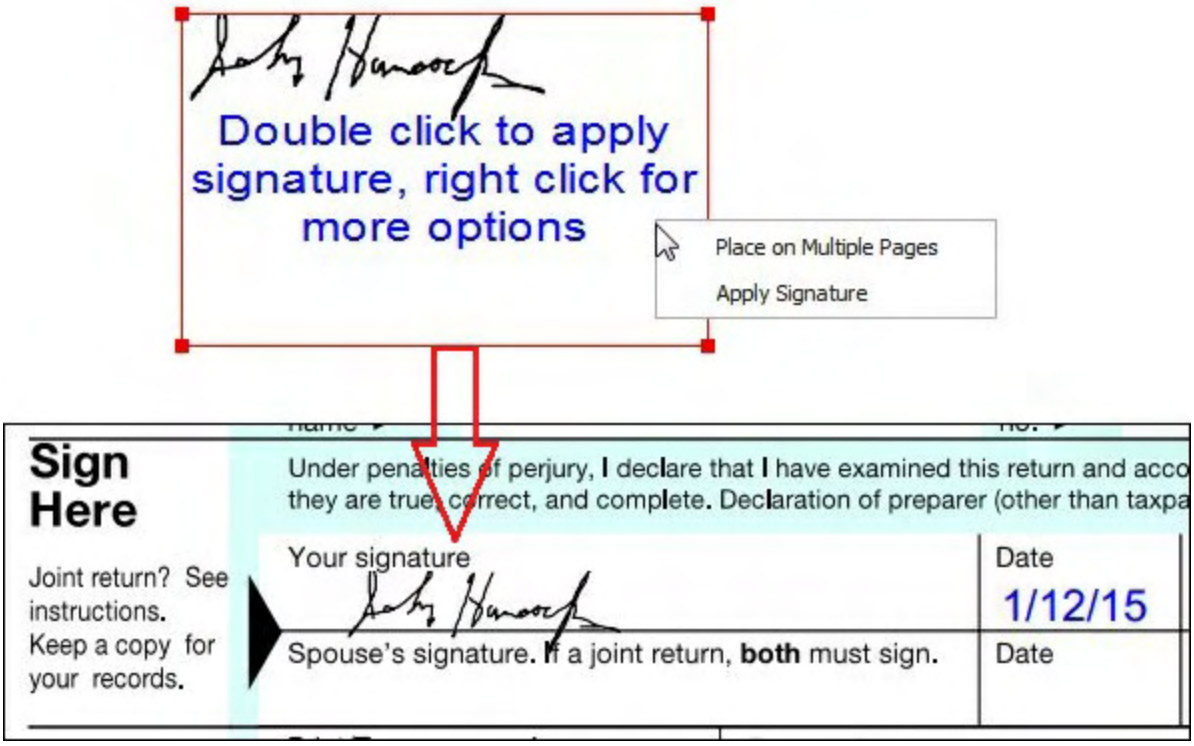


Figure 4. Select the signature, move and/or resize, then double-click when it's in the proper location.

If you want the signature to appear on every page, select Place on Multiple Pages from the right-click menu. You will still need to go to each page to locate and apply the image. This saves the time and paper it takes to print the signature page and scan it again.

Save the document and e-mail it back to the proper authority.

There is no real way to verify that it was actually you who signed the document. It still falls back on the old fashion notary public to properly verify signatures. While that face-to-face system is not perfect, it's still used for any documents that require an ironclad signature.

Whether signed by hand or electronically, the penalties for fraud are the same. When we enter into an agreement we trust that the other party will do so in good faith. As it turns out, your word is your bond. It's just a little easier to hold someone to their word if it's in writing.

While adding electronic signature may be a small step toward a paperless society, it's part of the cumulative effect. For now the scanner plays an important role in eliminating paper—at least until there is no more paper to scan.

Jack is the publisher of ComputorEdge Magazine. He's been with the magazine since first issue on May 16, 1983. Back then, it was called The Byte Buyer. His Web site is www.computoredge.com. He can be reached at ceeditor@computoredge.com. Jack is now in the process of updating and compiling his hundreds of articles and columns into e-books.

Currently available:

Recently released is Jack's FREE AutoHotkey book, AutoHotkey Tricks You Ought to Do with Window, available exclusively at ComputorEdge E-Books in the EPUB for e-readers and tablets, MOBI for Kindle, and PDF for printing formats.

ComputorEdge E-books is offering his AutoHotkey Applications, an idea-generating intermediate level e-book about using the AutoHotkey Graphical User Interface (GUI) command to write practical pop-up apps for your Windows computer. (It's not as hard as it sounds.)

Hidden Windows Tools for Protecting, Problem Solving and Troubleshooting Windows 8, Windows 7, Windows Vista, and Windows XP Computers.

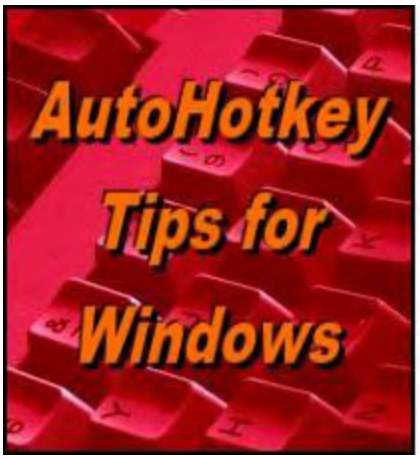
Jack's *A Beginner's Guide to AutoHotkey, Absolutely the Best Free Windows Utility Software Ever!: Create Power Tools for Windows XP, Windows Vista, Windows 7 and Windows 8* and *Digging Deeper Into AutoHotkey*.

Our second compilation of stupid *ComputorEdge* cartoons from 2011 and 2012 is now available at Amazon! *That Does Not Compute, Too! ComputorEdge Cartoons, Volume II: "Do You Like Windows 8 or Would You Prefer an Apple?"*

Special Free Offer at ComputorEdge E-Books! *Jack's Favorite Free Windows Programs: What They Are, What They Do, and How to Get Started!*.

Misunderstanding Windows 8: An Introduction, Orientation, and How-to for Windows 8 (Seventh Edition)!

Windows 7 Secrets Four-in-One E-Book Bundle,
Getting Started with Windows 7: An Introduction, Orientation, and How-to for Using Windows 7,
Sticking with Windows XP—or Not? Why You Should or Why You Should Not Upgrade to Windows 7,
and *That Does Not Compute!*, brilliantly drawn cartoons by Jim Whiting for really stupid gags by Jack about computers and the people who use them.

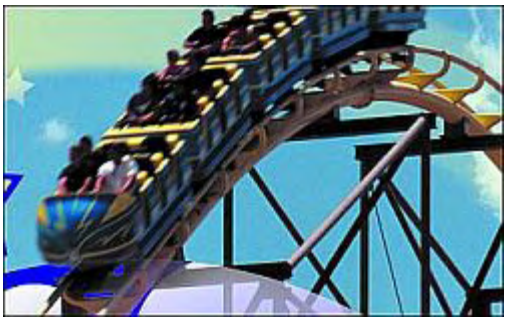


Understanding the Mysteries of Regular Expressions (RegEx) in AutoHotkey

"To Understand How a RegEx Works, It Helps to See Trains Running Down a Track"
by Jack Dunning

Many AutoHotkey script writers don't use Regular Expressions because they seem too mysterious and confusing. All they really need is a little understanding.

The use of Regular Expressions (commonly called RegEx or RegExp) in AutoHotkey is not a beginning topic and there certainly is nothing regular about Regular Expressions. I've spent the last few months exploring the programming tool and have developed a healthy respect for its flexibility and power. Many (including myself) have avoided using RegEx due to its enigmatic code which at times may appear almost incomprehensible. It's not like normal program code with *If-Then-Else* statements and *Loops*. Writing a RegEx is not merely a matter of following a logical sequence. It often requires a non-linear look at the problem. I've found that what helps me most is the analogy I maintain in my brain pan. That image gives me a basis for what a RegEx is trying to do. ("Try" is a good word when describing RegExs. Whereas the usual programming either works or doesn't work, RegEx "tries" to find pattern matches. If none are found, it moves on.)



Wikipedia describes a Regular Expression as "a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. 'find and replace'-like operations." I would describe RegEx as a data mining machine. RegEx is like a train rolling down a track of computer characters looking for patterns which match a specific set of given instructions. If it finds characters which match the pattern set, it grabs them and puts them on board the train.

As the RegEx train runs down the line, it continues picking up characters—as long as they fit the written instruction set. Some groups of characters may be saved for later reuse (back references). At times RegEx may look back at previous characters for validation or forward to coming data for confirmation (backward and forward assertions). While a particular RegEx may be forgiving in what it will accept on board, if the pattern does not completely match the given set of criteria, the entire group (including all previously collected characters) is kicked off the train and RegEx continues rolling along looking for the another possible set of matching characters. This continues until it either hits the ends of the line or finds a complete solution to its data schedule. Then RegEx stops. The RegEx data mining machine can be started up again by placing it in a Loop which restarts the same search for a point just beyond its current solution.

This is the image I visualize when working with a RegEx. The key to understanding RegEx is knowing what it is trying to do when it interprets the special symbols in a RegEx set of instructions.

Practical RegEx Uses

Maybe the most important question is, "If Regular Expressions can be so confusing, why bother?" Often when doing simple text searches or replacements it's quicker and easier to use functions built into a language. RegEx may be adding needless complication. However, a RegEx might do with one expression what takes several lines of code when using those other functions. It may take slightly longer (a few more microseconds), but the added flexibility could make the seemingly impossible a reality.

For example, IP addresses are varied and many although they all conform to a pattern. Each IP consists of four number (one to three digits long and between zero and 255) separated by a dot. With the proper RegEx, the engine can search through a document pulling out only the IP addresses. This was done in AutoHotkey to create a Web IP look up app to find the IP address geographic locations throughout the world.

Another use for a RegEx may be to find duplicate words in a document. This can be done with other functions, but it would take a few lines of code with conditionals (If-Then), whereas only one RegEx is needed. How about swapping the first and last words in selected text?

Maybe you want to strip all of the HTML code out of a Web page leaving only the text? Or, possibly you need to extract a list of all of the Web links found in a Web page? RegExs are the best way to ensure that a properly formatted, valid e-mail address is entered into a data field.

Pulling the numbers out of alphanumeric data is relatively simple with RegEx. Maybe a special symbols needs to be inserted in front of (or behind) each of a group of individual characters.

If it's a pattern you need to find (and possibly manipulate) in your haystack of data, then

RegEx may be your best bet. This may be all the incentive you need to explore the mysteries of Regular Expressions.

RegEx History

While not critical to understanding how to use Regular Expressions, knowing how such a strange programming animal came into being helps to develop a proper appreciation for it. The seeds of RegEx actually predate the first computer in mathematical recursion theory which "originated in the 1930s, with work of Kurt Gödel, Alonzo Church, Alan Turing, Stephen Kleene and Emil Post." But it wasn't until 1956 when Stephen Kleene first "described regular languages using his mathematical notation called regular sets." At this time, while computers did exist and were in use, they were still in their infancy.

In the late sixties, Regular Expressions came into popular use primarily in text editors. " Ken Thompson built Kleene's notation into the editor QED as a means to match patterns in text files." Two decades later, the pattern matching techniques in a more advanced RegEx library written by Henry Spencer became part of the Perl programming language. "Starting in 1997, Philip Hazel developed PCRE (Perl Compatible Regular Expressions), which attempts to closely mimic Perl's regular expression functionality…" PCRE is now considered the standard for RegEx engine implementation. The major programming languages now either include a RegEx engine or have a library implementation. While it may vary slightly, the symbolic coding is based upon PCRE. The AutoHotkey scripting language is no different and supports most of the same syntax and symbols as other implementation. Usually, example expressions found on the Web for other languages can be used directly in AutoHotkey.

There are a couple of RegEx symbols named after the mathematician Stephen Kleene. You may be familiar with the commonly used search and pattern match wildcard symbol * which is called the Kleene star. It's used to match none or more of a set of characters. In most computer uses we tend to use asterisk (*) to mean any character, but in a RegEx it applies to a previously matched set of characters while the dot (.) is the wildcard for any character. The Kleene plus + is similar to the star, but is used to match one or more, rather than none or more.

RegEx in AutoHotkey

There are two primary RegEx functions in AutoHotkey: RegExMatch() and RegExReplace(). While working with both over the past few months, it slowly dawned on me when to use each one. While there is some overlap, the two functions have distinctly different powers.

RegExMatch() is an extraction tool for finding and saving specific matches within the haystack of data. Once RegEx finds a match and extracts the data, the RegEx train stops in place. If you want it to keep going and find more matches to extract, then the function must be put inside a

Loop and restarted at its previous stopping point. *RegExMatch()* does not affect the original data while collecting matching data.

RegExReplace() is an alteration tool rearranging data as the train runs down the line. It finds a match, then replaces it with the appropriate changes. Unlike *RegExMatch()*, by default, *RegExReplace()* does not stop upon finding a match. In fact, unless told otherwise, the RegEx train continues locating the pattern and making changes until it reaches the end of the haystack.

Understanding the differences in these AutoHotkey functions makes it easier to pick the proper tool for the job. *RegExMatch()* is for mining, extracting, and replicating data ore for use elsewhere, while *RegExReplace()* is for changing and rearranging the ore inside the data mine.

The Best Way to Learn Regular Expressions

While I've written quite a bit about RegEx, reading about them alone is not the best way to learn. There is no substitute for writing an expression, then watching its effect. While there are many examples available either through the linked columns or on the Web, actually using them in an application will provide infinitely more insight into how they work. For AutoHotkey, I've found that Robert Ryan's Regular Expression Testing App is invaluable.

Ryan's Regular Expression Testing App

Originally when I started working with RegEx, I was writing tiny AutoHotkey scripts, testing my attempts, then making changes and testing again…and again., Eventually, I checked the AutoHotkey Scripts Forum and found this Regular Expression Tester written by Robert Ryan (see Figure 1). It's pretty slick, saves a lot of time, and offers instant understanding into how RegEx works.

It needs to be noted that this script was written for AutoHotkey_L and won't work with the Basic version of AutoHotkey. However, AutoHotkey_L is the currently accepted version of AutoHotkey and the download available on the site. Also, this RegEx tester was written for Perl Regular Expressions. While almost everything is identical when used with the AutoHotkey RegEx functions, there is one important difference. Since the functions delimit the RegEx with double quote marks, any double quote mark within the RegEx itself must be escaped with another double-quote ("") when used inside the AutoHotkey functions.

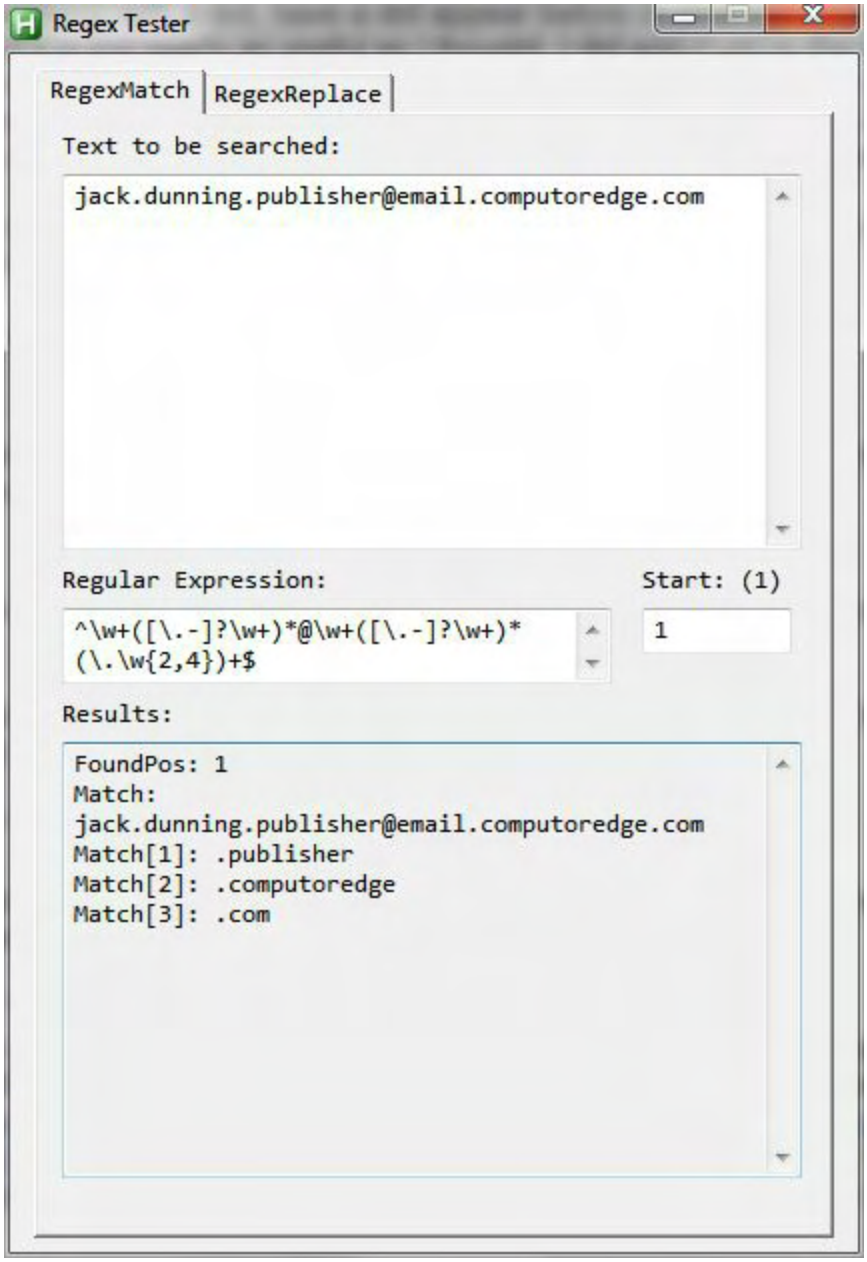


Figure 1. Ryan's Regular Expression Tester is used to evaluate the e-mail address validating expression used by the ComputorEdge E-mail Subscription Form.

This RegEx tester is interactive and immediately updates as either the input text data or expression is changed. For matching expressions (RegExMatch()—the first tab) the search string is entered into the top edit box. The RegEx is entered into the second edit box and the start character is entered into the next edit box (default is the first character). The results appear in the text box at the bottom of the window.

If the RegEx works and a match is found, then the results will show the match plus any subpattern matches (*Match[1]*, *Match[2]*, and *Match[3]*). If the RegEx fails to find a match, then *FoundPos* is 0 and *Match* is blank. The beauty of this tester is that you can change the data and instantly see if the RegEx is accepting or rejecting the data format or if there is a problem with the expression. Making minor changes can give you tremendous insight into how

RegEx works. It did for me!

In the course of writing the many AutoHotkey RegEx columns I used Ryan's Regular Expression Tester extensively and must give credit and pay homage to Robert Ryan who wrote the script. I've never met or communicated with him, but have benefited from his brilliance and one of the most useful programming tools I've found. Not only was I able to test my own expressions, but I was often able to change and simplify them based upon the instant feedback the Regular Expression Tester provided. Thanks, Robert! Wherever you are!

If you want to continue on with how to use RegEx in AutoHotkey, then my first introductory RegEx column was written for the beginning RegEx user and continues to build in the AutoHotkey columns in the following weeks.

* * *

Jack is the publisher of ComputorEdge Magazine. He's been with the magazine since first issue on May 16, 1983. Back then, it was called The Byte Buyer. His Web site is www.computoredge.com. He can be reached at ceeditor@computoredge.com. Jack is now in the process of updating and compiling his hundreds of articles and columns into e-books. Currently available:

Our second compilation of stupid *ComputorEdge* cartoons from 2011 and 2012 is now available at Amazon! *That Does Not Compute, Too! ComputorEdge Cartoons, Volume II: "Do You Like Windows 8 or Would You Prefer an Apple?"*

Special Free Offer at ComputorEdge E-Books! *Jack's Favorite Free Windows Programs: What They Are, What They Do, and How to Get Started!*.

Misunderstanding Windows 8: An Introduction, Orientation, and How-to for Windows 8 (Seventh Edition)!

Windows 7 Secrets Four-in-One E-Book Bundle,
Getting Started with Windows 7: An Introduction, Orientation, and How-to for Using Windows 7,
Sticking with Windows XP—or Not? Why You Should or Why You Should Not Upgrade to Windows 7,
and *That Does Not Compute!*, brilliantly drawn cartoons by Jim Whiting for really stupid gags by Jack about computers and the people who use them.

Wally Wang's Apple Farm

"Interesting Uses for Mobile Computing" by Wally Wang

Interesting Uses for Mobile Computing; Swift: The Future Programming Language of OS X and iOS; February's Apple Event; The Decline of Google Glass; Another Casualty of the Print World; Apple's Cocoa and Cocoa Touch Framework; The Services Menu.

When Apple introduced the iPhone and iPad, critics were quick to dismiss its importance—only to watch the entire smartphone and tablet market slavishly copy every feature of the iPhone and iPad. What makes smartphones and tablets so useful is that you can take a computer with you wherever you go.

The San Diego metropolitan transit system even offers iOS and Android apps so you can get real-time transit information from your smartphone. Now you won't have to guess how long you'll need to wait at a bus or trolley stop when your smartphone can give you an estimate. Would anyone ever want to access transit information using a netbook computer that's heavier and bulkier to carry?

By taking the power of computers away from desktops and laptops and putting them in your hands, mobile computing gives you the power to access computing power wherever you go. For blind people, download and use the Be My Eyes app for iOS. By using this app, blind people can aim the iPhone or iPad's camera at an item and ask a volunteer to tell what the camera sees.

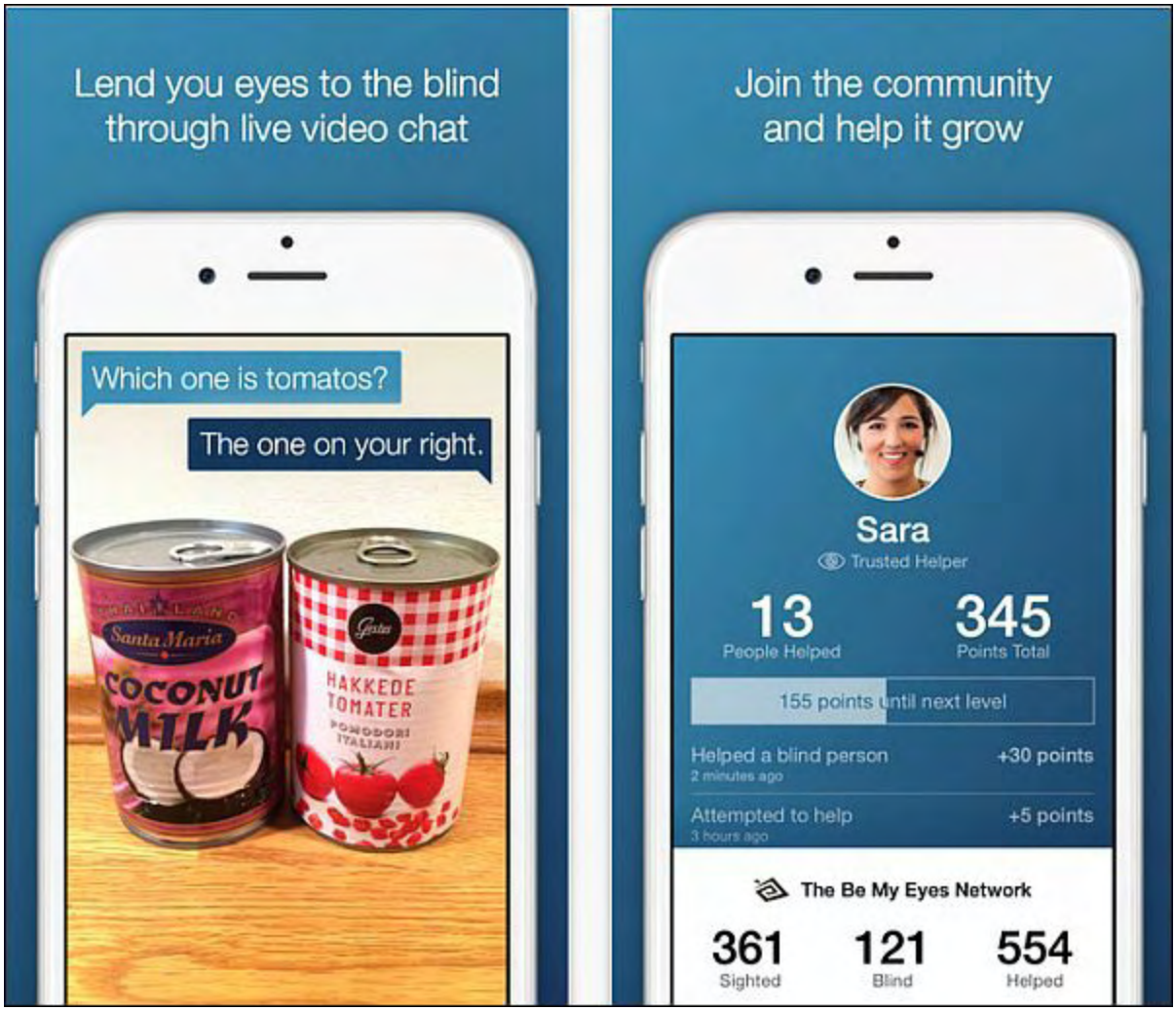


Figure 1. The Be My Eyes app helps blind people.

For example, suppose a blind person grabs a carton of milk and wants to know the expiration date. By pointing the camera at the milk carton and asking a volunteer connected to the Be My Eyes network, a blind person can get help from someone at all hours of the day. Anyone want to deny blind people access to mobile computing technology because they believe a netbook is superior to a smartphone or tablet because a netbook can run Adobe Flash, offer a USB port, or allow a replaceable battery?

The San Francisco Airport is testing the use of iBeacons to help travelers navigate their way through the airport terminal. When paired with Apple's VoiceOver technology, iBeacons can help blind people use their iPhone to tell them what's around. Now through iBeacon, they'll be able to find restrooms, restaurants, and even power outlets to recharge their iPhone.

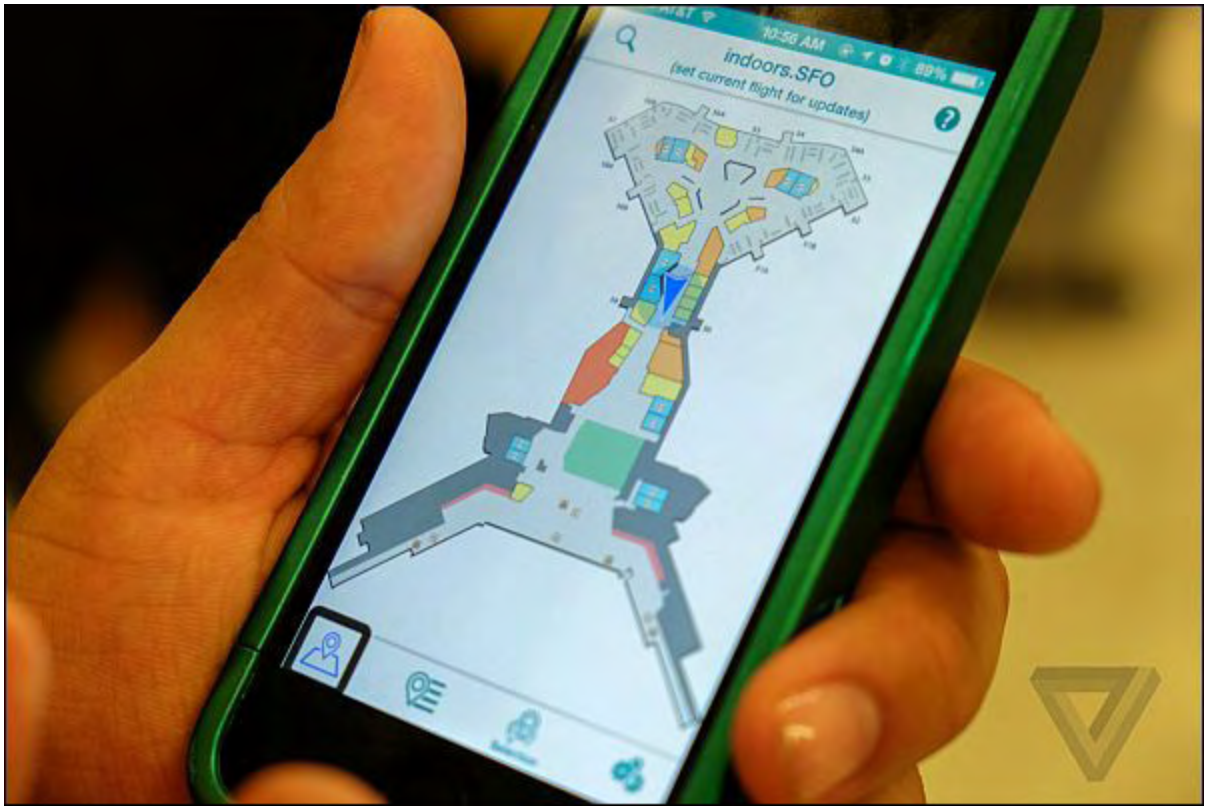


Figure 2. The San Francisco Airport will test iBeacons to help people find their way around.

The National Federation of the Blind and Ray Kurzweil, a noted artificial intelligence pioneer, have teamed up to create the KNFB Reader app, which lets blind people point their iPhone camera at printed text and have their iPhone read the text aloud to them. This gives blind people the ability to read restaurant menus, street signs, and other printed text that may not have any Braille equivalents nearby.

When you look at all the different ways that blind people can take advantage of mobile computers, you realize that mobile computing can benefit everyone (except those who insist that netbooks must still be superior because they offer a physical keyboard). Sometimes the real blind people are the ones who have the advantage of sight but fail to use it.

Swift: The Future Programming Language of OS X and iOS

When Apple announced their new programming language called Swift, it caught everyone by surprise. The main advantage of Swift is that it makes OS X and iOS development far easier than using Objective-C. Because the whole world is shifting towards mobile computing and mobile computing is still dominated by iOS, many programmers are simply following the money and switching their development efforts towards learning Swift.

Fortune reports that Swift's popularity has grown tremendously in the short time since it's

availability. The most popular programming languages continue to be JavaScript, C++, and Java, but Swift is rapidly catching up to Objective-C, which is mainly used to write OS X and iOS apps. Expect Swift to overtake Objective-C in a few years. Then expect Objective-C programmers to either switch to Swift or stubbornly resist the transition to Swift and gradually get left behind.

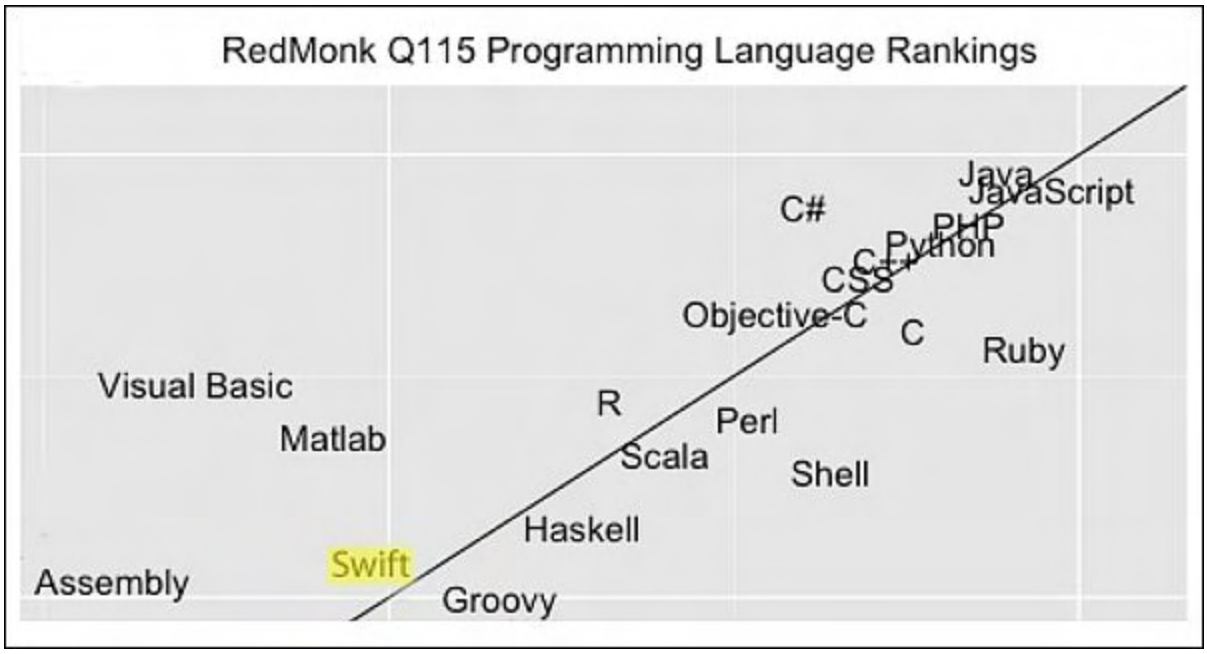


Figure 3. Swift is now one of the top 20 most popular programming languages.

If you're planning to write programs for a living, most college courses will focus on C#, C++, and Java with little attention paid to Swift. Unfortunately if you let college programming courses determine your future, you'll miss out on the future of OS X and iOS development, which means missing out on the future of mobile computing as well.

For programmers, you can't ignore the growing popularity of Swift, iOS, or OS X. There will still be plenty of programming opportunities for other platforms, but learning Swift should be your first priority if you're serious about staying current in the computing world. If you're not serious about keeping up with the future, just keep coming up with excuses for staying right where you're at. You'll always have plenty of company.

February's Apple Event

The latest rumor points to an Apple event sometimes near the end of February. Supposedly this next Apple event will highlight the Apple Watch and also introduce a low-end MacBook Pro with a Retina display to replace the current MacBook Pro line that still comes with an optical disk drive.

At this rumored Apple event, Apple should demonstrate the features of the Apple Watch so

people can see why they might want one. By that time, Apple should also officially release Xcode 6.2, which includes their WatchKit application programming interface (API) so you'll be able to write Apple Watch apps using Objective-C or Swift.

Just as the iPhone redefined the smartphone market and the iPad redefined the tablet market, expect the Apple Watch to redefine the wearable computer market. The idea of having a computer accessible at all times might seem like a trivial need, but critics thought the same thing when they dismissed smartphones and tablets because they believed desktop and laptop PCs could work just as well.

Anyone ever pull out their laptop from its carrying case, plug in a cellular modem, and turn the laptop on to check the weather, the latest sports scores, or the amount of time necessary to wait for the next bus or train? Now think how easy it will be to use a wearable computing instead of pulling out your iPhone or iPad to use it. 2015 should be an exciting year to see how creative minds find interesting ways to use the Apple Watch and usher in the next age of wearable computers to the world.

The Decline of Google Glass

Technology is only useful if you want to use it. That's the harsh lesson Google has learned with Google Glass, their wearable computer that cost too much ($1,500), looked weird, and scared everyone with privacy issues since they didn't want to be recorded. Gym locker rooms, movie theaters, and even some bars and restaurants quickly banned anyone wearing Google Glass. Now Google has scaled back their efforts.

Most likely Google Glass will never become a mainstream product but can become a niche product for police officers, firefighters, soldiers, and other people who may have a specific need to record video of their actions. As a niche device, Google Glass should still thrive. As an ordinary product for consumers, expect Google Glass to fade away.

After all, if a wearable computer makes you look weird and makes people uncomfortable around you, it doesn't matter how good the computer may be if you can't use it when and where you want. Unlike Google Glass, the Apple Watch is meant to be stylish so owners will want to show it off. Also unlike Google Glass, the Apple Watch won't threaten anyone's privacy to make them uncomfortable.

The technology behind Google Glass may be interesting, but until Google finds a way to make a wearable computer fun for people to wear and acceptable for the public to allow, then Google Glass will likely go down as an interesting technological footnote in history. As Google Glass learned the hard way, sometimes technology is more than just a laundry list of technical specifications. Sometimes you actually have to consider people's emotions as well, which is something tech-savvy users often ignore as they salivate over the latest and longest list of technical features that don't always translate into anything useful.

Another Casualty of the Print World

Even the most die-hard, anti-Apple critic can't ignore the influence the iPhone and iPad has had on the computing market today. Yet despite the growing influence and profitability of Apple, another Apple-related magazine has decided to shut down. MacUser, launched in 1985, has announced they will shut down after their February edition. This occurs just several months after MacWorld shut down their print publication as well.

With Apple selling more Macintosh computers than ever before along with iPhones and iPads with the Apple Watch ready to go on sale soon, how come two major Apple-related magazines can't stay in business? The answer isn't that the magazines aren't covering popular topics, but that magazines are too expensive to print, ship, and store when it's so much easier, faster, and more timely just to update a Web site instead.

If magazines can't survive and newspapers are still struggling with printing yesterday's news in less detail than Web sites can do, printed books are the next to continue fading away. With Borders Books gone and Barnes & Noble struggling, printed books are definitely not the future any more. Just look at the computer section of any bookstore and you'll see the number of titles has shrunk dramatically.

That's because information keeps changing so rapidly that keeping a printed computer book on the shelves no longer makes sense when its information becomes outdated a few months later. In the old days, computer information changed slowly so computer books had a longer shelf life. Today, computer books offer outdated information just like newspapers, so why bother wasting time and money with printed books when you can find answers on the Internet instead?

By simply seeing the obvious, anyone could have anticipated the slow down in the world of print. By angrily denying trends right in front of their face, too many people ignored the facts before suddenly realizing that the world of print is gone for good. When people ignore the obvious and then suffer later, you have to wonder if they'll ever learn the next time ominous trends threaten their lifestyle. (The answer is probably no.)

Apple's Cocoa and Cocoa Touch Framework

Microsoft learned that the best way to insure their dominance was to offer operating system interfaces that would make it easy to create Windows programs. In the process of making it easy to create Windows programs using frameworks like DirectX, Microsoft also made it difficult for programmers to port programs to other operating systems.

Now Apple is using that same technique with their Cocoa and Cocoa Touch frameworks for creating OS X and iOS apps respectively. By using Cocoa/Cocoa Touch, you can create

sophisticated programs that include the operating system's built-in spell checker, iCloud access, and undo commands. If you choose to create a cross-platform program, you'll have to add these features in yourself, which means writing a lot of extra code to provide standard operating system features.

Of course, once you start relying on Cocoa/Cocoa Touch, you won't be able to port your program easily to another platform. So the trade-off is to rely on a framework to help you write a program easier and faster, but can't be ported easily. Or you can write a program that can be ported easily but which requires you to add your own standard features, which will take longer and require additional testing.

For optimum speed and efficiency, it's best to rely on specific operating system frameworks. For maximum flexibility and portability, avoid any specific operating system frameworks.

Given the growing dominance of iOS in the mobile computing market, you probably can't go wrong focusing on iOS and relying on the Cocoa Touch framework while using Swift or Objective-C to create your app. It may not be portable to other operating systems, but if you can create an iOS app quickly and easily for the most lucrative mobile market, then you have to decide how much other mobile platforms really matter.

The developer of a game called Monument Valley lists their revenue from Apple's App Store (81.7%), Google Play (13.9%), and Amazon (4.3%). Although Android dominates the mobile operating system market, that large number never seems to translate into more sales and higher revenue for mobile app developers.

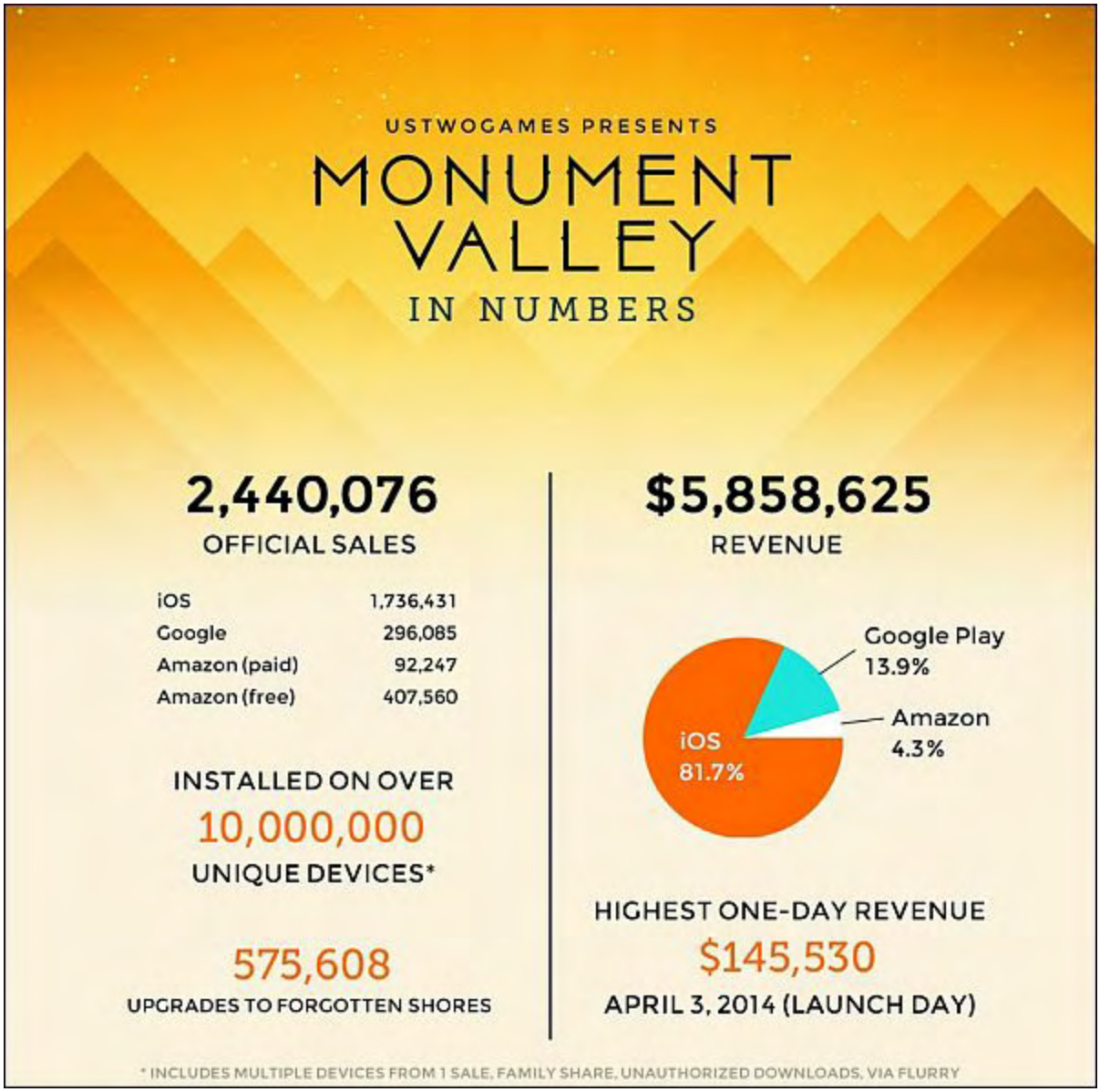


Figure 4. The percentage of revenue coming from iOS dwarfs other operating systems.

With the choice between the most lucrative market (iOS) or a larger market that generates far less income (Android), it's easy to see why developers would want to target iOS initially.

* * *

Open almost any program, select text, and click on its menu such as the Word menu in Microsoft Word or the Pages menu in Pages. Now choose Services and you'll see a variety of ways you can send, save, or share your selected content.

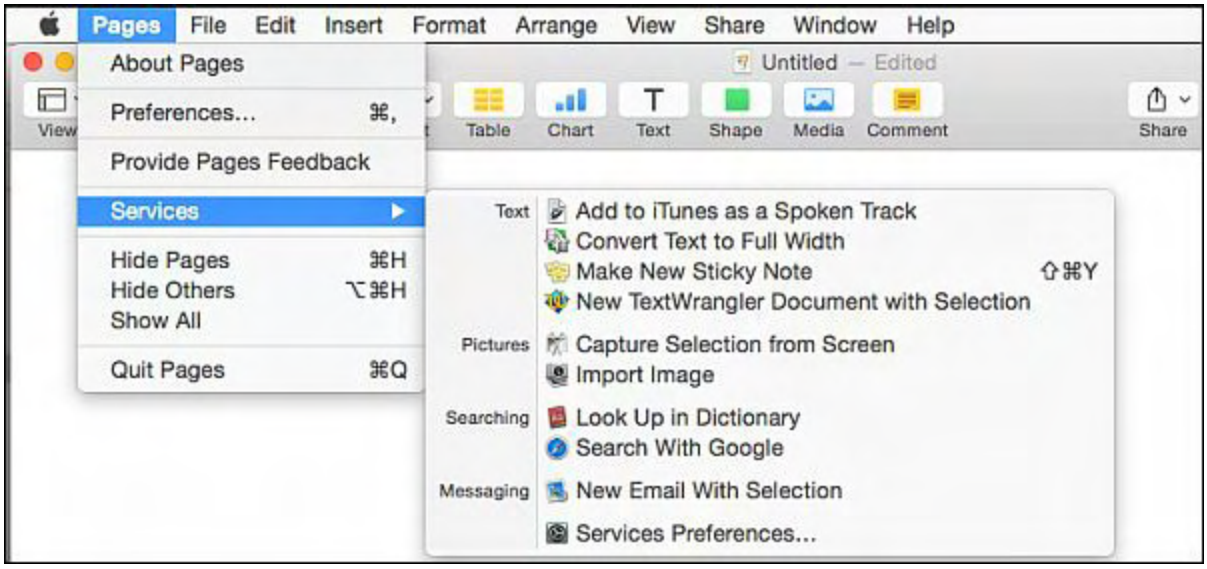


Figure 5. The Services menu gives you options for sharing selected content within almost any program.

In the early days, before Wally became an Internationally renowned comedian, computer book writer, and generally cool guy, Wally Wang used to hang around The Byte Buyer dangling participles with Jack Dunning and go to the gym to pump iron with Dan Gookin.

Wally is responsible for the following books:

Microsoft Office 2013 For Dummies

Beginning Programming for Dummies

Beginning Programming All-in-One Reference for Dummies

Breaking Into Acting for Dummies with Larry Garrison

Strategic Entrepreneurism with Jon and Gerald Fisher

How to Live with a Cat (When You Really Don't Want To)

The Secrets of the Wall Street Stock Traders

Mac Programming For Absolute Beginners

Republican Fairy Tales (Children's Stories the 1% Tell About the Rest of Us)

The Zen of Effortless Selling with Moe Abdou

The 15-Minute Movie Method

Math for the Zombie Apocalypse

How to Write a Great Script with Final Draft 9

Making a Scene: The Science of Scene Structure.

In his spare time, Wally likes blogging about movies and writing screenplays at his site "The 15 Minute Movie Method," finding interesting news stories about cats at his site "Cat Daily News," giving advice to authors who want to self-publish e-books at his site "The Electronic Author," and providing the type of advice he wishes someone would have told him when he was much younger at his personal Web site. Wally can be reached at *wally@computoredge.com* or you can follow him on Twitter @wallacewang_com.

Editor's Letters: Tips and Thoughts from Readers

"Computer and Internet tips, plus comments on the articles and columns." by ComputorEdge Staff

"Backing up Files with Long Names/Paths," "Internet over Power Lines," "Scan with HP Officejet 8600," "Micro Power Now! (Circa 1983) Comments"

Backing up Files with Long Names/Paths

[Regarding the January 2 Digital Dave column:]

Let me add another program suggestion that does not balk at long names or paths: Folderclone from Salty Brine. I've used it for several years now, and find it very user-friendly for this very non-techie user. :)

-Jan, Murrieta, CA

Internet over Power Lines

[Regarding the January 2 Digital Dave column:]

Some years ago several cities tried Internet and/or TV over power lines. The power lines radiated so much noise it precluded operation in some amateur radio bands. I suspect it did with other services also. Optical fiber and coax cable keep the signals contained within, therefore no radiated signal. In addition to economics, the noise problem also contributed to its demise.

-Chuck Banks W6HYY, Oakhurst, CA

Scan with HP Officejet 8600

[Regarding the January 2 Digital Dave column:]

In answer to Steve's inquiry on scanning from the 8600. I tried using the printer panel once for scanning and didn't like it, instead I use the HP supplied Scanning Wizard which gives you the save to option for your documents.

-Richard DiSanto, Denver, CO

Re: Steve who had a problem scanning to a location other than his "Pictures" folder. In your answer you discussed scanning from the "control panel" on the printer. I have an Officejet 8610; maybe Steve's printer is different, but I have no option to scan to computer from the printer's control panel. What Steve needs to do is to use the software that came with the printer; in the scan options there should be options such as scan to PDF, scan to OCR, scan to JPEG, etc. Whichever option he chooses, there should be an "Advanced Settings" link; after clicking that link, there should be a "Destination" tab, and on that tab there is a "Save Location" dropdown.

-Dan Jacobs, San Diego

Micro Power Now! (Circa 1983) Comments

[Regarding Jack Dunning's January 2 article, "Micro Power Now! (Circa 1983)":]

That was a pretty good vision of our current technology back then Mr. Dunning. I had just started at Miramar College and got my degree in Computer Science. I remember in one of my Pascal classes, the instructor told me if I were to get into anything, it should be graphics. He was right. I've lived in VA since 2005, but I'm a San Diego native. I used to get the *The Byte Buyer* since it came out. I get the *ComputorEdge* now in my e-mail whenever you started making that available. Great job with the magazine and much continued success. Thanks for all the information. I've used much of it to my advantage.

Happy New Year!

-Tom Zamora, Midlothian, VA

My first computer was a 1983 Actrix DS. Check it out here: Actrix DS

-Kay Katz, San Diego, CA

ComputorEdge always wants to hear from you, our readers. If you have specific comments about one of our articles, please click the "Tell us what you think about this article!" link at the top or bottom of the specific article/column at ComputorEdge.com. Your comments will be attached to the column and may appear at a later time in the "Editor's Letters" section.If you want to submit a short "ComputorQuick Review", or yell at us, please e-mail us at ceeditor@computoredge.com. If you would like to review our recent e-books, please visit ComputorEdge E-Books.

Send e-mail to ceeditor@computoredge.com with questions about editorial content.

Table of Contents

List of ComputorEdge Sponsors 2
San Diego ComputorEdge Sponsors 2
Colorado ComputorEdge Sponsors 2
ComputorEdge™ Online — 01/23/15 3
Signing Digital Documents 3
Magazine Summary 3
Digital Dave 5
Password Management Programs? 5
How to Keep a Safe Password List 6
Dream Up a Master Root 6
Create a Password Extension for Each Site and Program 7
Trivial Web Site Accounts 7
More Safe Password Tips 8
Scanning Makes Us Paperless--Almost 9
Not Paperless, But Less Paper 9
Enter the Scanner 10
Print, Sign, and Scan 11
Electronic Signatures 11
Using PDF Sign to Save Paper 12
Understanding the Mysteries of Regular Expressions (RegEx) in AutoHotkey 17
Practical RegEx Uses 18
RegEx History 19
RegEx in AutoHotkey 19
The Best Way to Learn Regular Expressions 20
Ryan's Regular Expression Testing App 20
Wally Wang's Apple Farm 24
Swift: The Future Programming Language of OS X and iOS 26
February's Apple Event 27
The Decline of Google Glass 28
Another Casualty of the Print World 29
Apple's Cocoa and Cocoa Touch Framework 29
Editor's Letters: Tips and Thoughts from Readers 34

Backing up Files with Long Names/Paths 34
Internet over Power Lines 34
Scan with HP Officejet 8600 34
Micro Power Now! (Circa 1983) Comments 35