



Apple Blossom Computer Club
A registered Apple/Macintosh User Group



Aug '08

still only

\$2.00

The

ROSE



BYTER

Next Meeting

August 21, 7 PM

Creative Images

Printing

2042 NE Airport Rd

541-673-6098

Agenda

1. See the very latest in printing technology.
2. Whatever else strikes you at the time.

<http://www.creativeimages.net/>

Keeping In Touch

by Jim McClellan
<mcclellan@charter.net>

As some of you may know I have been a member of Apple Blossom since 1981. At that time we all had Apple II computers; the Mac hadn't come out yet. This meant that virtually everything we did with our computer was via inputting text, not clicking on graphic images with a mouse.

Several years later I bought a 300 baud modem that would let me communicate with other folks who also had a modem. This also was via text. I remember going "on-line" to have a "chat" with a friend who lived down in Phoenix, Oregon. We were chatting about tourism things and suddenly someone from some other location in the USA was "on-line" with

8 -->

YOUTUBYTER

by Dave Archer <dave@davearcher.com>



Dateline: Youtubyter - August '08

---Sea Angels

Don't miss below, the Youtube of these critters!!! Slowly beating their parapodia, sea angels gracefully fly through the upper 20 meters of the water column. Although usually slow-moving, they are capable of surprising bursts of speed. Shallow water animals are proving to be as amazing as the deep ones.

<http://www.youtube.com/watch?v=vB5recdpPaI>

--- "Reincarnation,"

A poem written by Wallace Mcrae, aka, "The Cowboy Curmudgeon" is a poem

2 -->

The **Apple Blossom Computer Club** (ABCC) is an Apple Computer Inc., registered Macintosh and Apple][family user group. The ABCC publishes *The RoseByter* newsletter monthly which is posted to each paid up member and reciprocating user groups. ABCC participates in user group newsletter content exchange. The ABCC also maintains a WWW site at:

<http://www.abccmug.org>

Membership

Just \$20/year! Send with your name, snail- & e-mail address & phone to:
ABCC
13748 Lookingglass Rd.
Winston, OR 97496

Current ABCC Leadership

Treasurer

Jim McClellan
<mcclellan@charter.net>

Apple Ambassador

Jim McClellan

Web Master

Jim McClellan

AppleScript Guru

Jack Webster <jackw@rio.com>

Newsletter Editor

Walt Pawley <walt@wump.org>

Send your stories and newsletter ideas to the Editor, Walt Pawley, at <walt@wump.org>. Plain text files are preferred, sent within the body of an email message or as an attachment. Mail physical media to:

**676 River Bend Road
Roseburg, OR 97470**

Please understand that materials submitted may not be used and those that are will likely be edited.

Copyright© 2008, All Rights Reserved



World Famous Chef Hard at Work!

<-1 YOUTUBYTE

that my brother has recited at barbecues for decades. This recitation is by Gene Perkins at the Southwest Bluegrass Club in Grapevine, Texas. There are other versions on Youtube. I have yet to find Wallace Mcrae reciting his own poem.

<http://youtube.com/watch?v=MNwz9DVgAPs>

--- Finland Army Secret Weapon Jet Boots!!!

I don't know if this is really a Finland Army Secret Weapon, or the person who posted it was making a "Finn joke". This guy tests out jets on his boots, then, wearing a simple costume like a flying squirrel, jumps out of an airplane, and jets around in ways you won't believe.

<http://www.youtube.com/watch?v=OVxeZYdVRLM>



Dave Archer -- <http://www.davearcher.com> (541) 440-0002
Dave Archer Studios, 1541 Buckhorn Road, Roseburg, OR 97470
SEE DAVE ARCHER STORE

<http://www.cafepress.com/davearcher>

Lawyers Are Killing Us

Perhaps even literally. But it's our own fault, of course. We let them. Indeed, we even demand it.

One example jumped on me recently. I wanted to get some 5.25" external drive enclosures. I like them because they can accommodate both hard disks

and CD/DVD drives and they have internal power supplies—no power glob required. Or, at least, they used to. Not any more. They are too dangerous for consumers to handle. They might get shocked and sue. Soon the only product acceptable may be an indestructible, hypoallergenic padded room to sit in. Ed.



Can You Unravel the Coupon Code?

One sure way to do it is to come to the meeting and win the door prize.

The good folks at MacProVideo produce some great learning and reference videos as several ABCC'ers have discovered. Here's your chance to get a \$50 product of your choice from MacProVideo.

<http://www.macprovideo.com/>

Age Old Data

As we age, more of us encounter difficulties with dealing with old data. I've been asked about how to deal with the issue by people recently so I thought I'd share some thoughts from my cranial trove of misinformation.

Carrying data forward into time is not really all that easy ... if one cares to use it after a long period. There are basically two reasons: readers for the media frequently won't work or exist in the future and formats from long ago are also long forgotten. I've experienced both these issues many times.

For example, I have two boxes of 80 column "IBM cards" in my mess around here. Decades ago they became obsolete. The equipment to deal with them faded away. I wanted to get the data from these cards to some "modern" media while I still could: a 2400' reel of 1/2", eight track magnetic tape... also now used almost not at all. Later, I wanted to access some of this data. Created on a Control Data Corporation 3300 computer in COSY format, I was able to write a program for the computers I was using at the time to "understand" it.

Another example is Apple formatted diskettes. There's little question in the mind of anyone technically competent that Steve Wozniak's data handling on floppies was far superior to the nonsense used by IBM PC diskettes. Such things do not matter to the marketplace, which gladly adopts inferior products if the hype is adequate. Time and again, Apple has ended up adopting IBM PC style technology despite having better designs in the field. We really are our own worst enemy.

So... how can you hedge your bets on keeping your data over the long term?

If anything, it has to do with using data formats that are open and common. By being an open standard, you should be able to get a description of how the format works that's adequate to write a program to decode it. By being common, there will be more

by Walt Pawley <walt@wump.org>

likelihood that the format will survive for a long period.

One might think that using Microsoft products would be a good hedge. After all, it's used by more people than anything else. Like almost all vendors of proprietary products, Microsoft changes it's formats routinely and quits supporting the translation of old formats into new on schedule. Open Source vendors do similar things but the difference is that their formats are documented. That description is most commonly coded in 7-bit ASCII (American Standard Code for Information Interchange). ASCII is almost universally interpretable on computers and has been for a long time. Thus, it's best, even if you're Chinese living in China, to stick with 7-bit ASCII from the archival perspective.

I'm not that fond of so-called word processing programs. They basically attempt to do what page layout software does but with peculiar limitations and proprietary formats. If you insist on having fancy fonts, multiple sizes, etc. in a fundamentally textual document, I'd suggest exporting to RTF (Rich Text Format) for archival purposes. Worse coming to worst, the ASCII text can be extracted from RTF.

PDF (Page Description Format) is supposed to be a "standard." I guess it is ... if you don't mind it changing every time Adobe wants a money shot. While the more modern form of PDFs seem more compressed than earlier ones, there's this little problem with reading them: many programs don't know how. If you can, I suggest saving files in PDF 1.3, for example. It's widely, if not quite universally, readable. I recommend *not* using PDF for archival purposes.

Formats for images are problematic as well. If there's an advantage, it's that many of the formats available are well described in openly available publications. There's much to consider: size, image quality, likelihood of reconstruction in the face of data loss, etc.

I'd avoid archiving images in any proprietary format, including Photoshop's. JPEG, image quality permitting, is likely to be long supported, it's well documented and used by all kinds of things, like cameras. Barring some magic to do a much better job of image compression, JPEG will likely be with us for a long time.

Video files are logically more complex and very much larger. The problem with archiving video is that there are just way too many alternatives. Worse, competing business interests, have created their own "standards." You can get programs that translate amongst numerous formats (VisualHub is one; it uses open source code underneath its GUI). If I had to choose just one format, I suppose I'd choose MPEG.

Programs are something you should not expect to be able to use in the future, though many have remained functional by some means long after no direct hardware is available. But this is unlikely to continue for the long term. Old computers were simple and, therefore, easy to emulate. I'd seriously question that designers of today's processor chips are certain they really work, much less that some emulated form will run any program that would run on the silicon it was intended for. Even keeping old computers has its limits: parts fail. Parts, once common, frequently have no modern counterpart today. Modern computers are far more problematic since they are largely composed of proprietary chips used in a single run of a computer model and computer models are changing sub-annually these days.

What does all this mean?

I guess it mostly means that there's nothing even remotely close to a panacea available. My advice is to try to prioritize the importance of your data. This means work (ugh). Export that data into formats that are likely to have a long life and have an open definition.



Water Blogged Wump

Any trace of organization in these paragraphs is entirely coincidental

4



@ What the He... Museum

Welcome to the Silicon Desert! I suppose one might consider that some sort of “clever” play on words, what with deserts being composed largely of sand and rock, which tend to be just loaded with silicon, but that’s not my intent. You see, it seems that all that money we spend on oil is now being used to create artificial oases, **one of which goes by the moniker DSO** (Dubai Silicon Oasis), as attractants for creators of modern technology. The DSO is being raised out of the desert next to the ocean by several billion dollars worth of infrastructure development that is just for the creature comfort of those being attracted. If they decide that the DSO is a good place to operate their business, they’ll have to bring their own operational capital along for the ride. But they and their minions will have ready made suburbs to live in and can work on their tans while not beavering away at their workbenches and in offices. They even intend to pop up a world class university right there. I suppose this whole business makes as much sense as anything these days. For me, the real question is whether things like DSO will have their effect on the politics of the region or not. It’s hard for me to imagine building good quality nanotech stuff with terrorist bombs rocking the neighborhood.

There’s a techno-epidemic in progress and no one even knows about it. Or, at least, those who do know about it are keeping it pretty close to the vest. **I’ll give it a name — capacitor cascade failure syndrome. CCFS** is a designed-in way for systems to crap out and is mostly driven by pushing

the design envelope too hard. These days, one uses a lot of small capacitors in parallel to provide the effect (more or less) of what used to be done with one large capacitor. There are several reasons why this is done: the sum of cost of the small parts is less, the small parts can be automatically assembled to boards, the small parts can be fit into oddly shaped places, the small parts can provide better electrical performance and probably a bunch more. This form of design has been going on for some time. Fueled by the need to provide ever increasing amounts of clean power to the processors, better designs for the capacitors have been churning out in droves. The trouble is that designers of equipment believe they can use the specifications given to these capacitors. Thus they only put in just as few as the specs indicate will do the job. They forget to ask the question: what happens if one of these capacitors fail? The answer is simple; the rest of them are overloaded and fail. Apple and some of their customers fell victim to CCFS with the early iMac G5’s in a big way. First, Nichicon made some capacitors that didn’t meet the specifications they were supposed to meet (not that it really would make much difference — it just speeds things up). Second, Apple split the capacitance used between the main board and the power supply proper. This meant that when CCFS occurred, both items died. Out of warranty, it’s a \$700 repair job. In the PC world, no one cares. Why? Well, the motherboards and power supplies are throwaways.

Almost everyone knows that electronics keeps bringing us more and more capability in smaller, less expensive packages almost daily. What fewer of us are aware of is the basic reason why. They’ve probably been exposed to it repeatedly, in statements about the numbers of transistors that can be put on a chip, however. The reason for

this progress is what has seemed to be a never ending shrinkage of the size of stuff that can be manipulated with care. The battle has centered around how to do ever more precise lithography, ie. printing. This is because the manufacture of chips revolves around creating patterns of chemicals (the “inks”) on the surface of wafers — 300 mm diameter disks of very carefully made rock (the paper). That capacity is reaching something of a practical limit at 45 nanometer line widths. The chip industry currently does not have a viable means of moving to 32 nm, which is touted as the next step on the way. It’s not so much that there is no way to do it. It’s more that it’s extremely expensive to get in a position to try and, once there, very difficult to get to a point of knowing how to get consistent, useful results. This is because the number of atoms spanning 32 nm is something like 100. One can see that **at 300 picometers we have a hard limit** — unless we can somehow figure out how to shrink atoms ... without also changing their electrical properties. Long before such considerations come into play, the mere lack of enough atoms in the structures being “printed” will stop the process because many of the effects utilized in circuitry require some bulk in the materials. Even “single atomic layer” devices generally use many layers to perform their magic. Another way to think about this whole business is that we once again live in interesting times — the transition from steady progress in a straight line to that state where we don’t see any means of improvement. It could be exciting to see whether there are segues in the offing.

I’ve long been annoyed by the ceaseless orphaning of old software, a prevalent process which seems to be getting worse. Of course, I’m accustomed to people telling me how Apple has done wonders keeping old

5 -->

<--Water Blogged Wump

software working in their new operating systems. But times are changing rapidly. Almost everywhere you look, you find “old” versions of programs will run on Tiger (Mac OS X.4) or later. And, if they happen to have a legacy version that will run on earlier versions of Mac OS X, chances are the new version’s file formats won’t be compatible with the old version’s. I thought the following (from George Warner, Apple Developer Technical Support) might be indicative of what’s going on: “We typically try to respond within the initial 48 hours. **We are iSwamped right now** (most of us have 12+ incidents in our queue; this is way above normal for us).” There may well be some marvelous goal Apple is attempting to accomplish. But I’d much rather they would quit breaking things for users in the process. I’ve not noticed anything all that compelling that’s bubbled up out of the morass. People need time to digest what they have and then they need to be able to use it for their purposes. Continual “upgrading” may be a nice game for some minority of users but my guess is that that’s not most of us. Apple is not alone in doing this to their customers. Microsoft is on that gravy train even though they really don’t need to be (they have a self-captive audience). Even Linux and FreeBSD are in a rapid state of churn. I suspect the latter is in large part in response to Apple’s constant pushing. If you’ve not heard of femtocells before, as I had not, here’s a chance to get mildly acquainted. I was stumbling along in a techno-rag this evening when I stumbled on the term. I was rather expecting it refer to some sub-nanometer sized atomic wizenoid. As I read into the article, however, I discovered it has something to do with 3G cellular phones. It seems that 3G cellular phones are a hardship on cellular phone service providers, so **they’ve come up with a really great idea: femtocells**. The concept is simple. They sell you a femtocell device

that you hook up to your broadband Internet connection and it serves as a local cellular phone “tower” right there in your home. Not a bad deal, eh. You buy the equipment, pay for the alternate communication path and the electricity it takes to run it and they charge you for the minutes you use. Maybe “deal” is the the wrong word. Perhaps “racket” is closer to reality. Of course, you still get to use your cell phone in that good old fashioned way too but they may want to charge you more for that. I ran across some other things in the article that I thought you might like to know. For example, it says that 53 percent of American homes now have broadband connections to the Internet which compares to 89 percent in South Korea (ain’t free enterprise just grand). It also said that US consumers logged 1.1 trillion connection minutes in the first half of 2007, with a growth rate of 63 percent per year. That’s 100 hours per year for every person in the country and a lot of us don’t have cellular phones (like me, for example).

Dovetailing into the last paragraph (quite by accident, I assure you), it seems that some major Internet Service Providers (ISPs) are tired of their meager profits. **They like the nickle-and-dime-you-to-death billing method**. You know ... they provide crappy service at a low speed for a fixed price for some small number of bytes of data transfer. When you exceed that threshold, the meter starts running and you pay extra for the “excess.” It’s certainly great to have competition in that good old Murkin way. You know, the way we have of splitting up infrastructure so that there isn’t any. I’m not really a big fan of free enterprise and competition as a means to operate infrastructure. True, it’s a great excuse to get megalomaniacs to steal as much capital as they can and run roughshod over whatever gets in their way. That way, when you tout up the cost of creating some infrastructure, you get to ignore all the loss the losers put into their megalomaniacs. The real prob-

lem with this scenario is that when the dust starts to settle, megalomaniacs have to do something else to satisfy their desires, so they start the nickle-and-dime crap. It’s bad enough that ISPs don’t really provide a network. Instead, they provide trees — your packets must travel up down the branch your connection is on, moving closer and closer to the trunk. When they get to the trunk, they get passed to the trunk of some other tree, where they move back out toward the branch of that tree you’re communicating with. While something of an over simplification, this characterization of Internet connections is largely accurate. It’s also very much unlike what ARPA had in mind when the Internet was first conceived. The reason is simple - profit. Perhaps one day humans will be able to create and operate infrastructure in rational ways. I’m not holding my breath.

I can recall the days of my youth when the IBM 360/50 was considered something of a “large scale” computer system. In particular, I remember visiting a “credit bureau” in Portland that had one. The age of computers keeping data on people had really just begun. I did not like what I saw then. Well, it’s much more advanced these days and, **despite all the pleading we’ve whimpered over at least the last 50 years**, much more invasive and unfair than ever. I got an email today titled “Toor2122 - Steve Rambam - Privacy Is Dead - Get Over It,” which pointed to a nearly two hour long movie hosted by Google: <<http://video.google.com/videoplay?docid=-383709537384528624&pr=goog-sl>>. I strongly recommend sitting through a good portion of this presentation. It’s not particularly technical. If you do, perhaps you can help me answer the question of what can be done about it. We could leave things as they are, which allows people with money to spend, to use data about you against you (including crooks). Or, we could make crooks out of the people who accumulate, buy or sell such data. Or, we simply

<--Water Blogged Wump

demand that all such data become publicly and freely available as a matter of course. The latter sounds like a horrible idea ... if you're one of those people who still harbor the illusion that you have some sort of private life. If the rest of our existence is any guide, making crooks out of those who traffic in such data would simply raise the price of that data. The current situation is only not completely horrific because of the societal momentum against the use of such data. That momentum is winding down and will have dissipated almost completely in another generation. Do you have a good idea about how to deal with this?

Now for a problem we could actually do something about. I suspect, that like most of the ills I perceive, even though something could be done about it, we'll manage to avoid that at all costs. You see, I really love leaves. With the increasing population density surrounding me, they're more important to me than ever. Why? It's my old pet peeve: light pollution. Of course, it's just me. **I'm quite puzzled by most people's attitude about lighting everything up.** They do so, for example, to increase their belief that they are more secure at night. For example, we had a rash of surreptitious wee-hour-of-the-morning activity in our neighborhood quite some time ago. A few days later, a robbery ring that had been working the area was arrested. One of our neighbor's response to this was to add a bright, automatic yard light to their house. Now, any crook wanting to walk around there doesn't need to carry a flashlight. Unfortunately for us, the neighbor's light shines all over parts of our yard. But this time of year, the leaves on the bushes and the trees make the situation tolerable. The leaves also block almost all the more distant yard lights that other people think are keeping them secure. So, when I wander out back on a nice, partially overcast late night, I see the hill across the river lit up by

the lights from the sawmill that's up the river from us and there's more than enough light from the reflection of Roseburg off the the clouds for me to see my way around. About the only time it's nearly dark here is when it's a new Moon on a nearly clear night. For some reason which escapes me, we don't think of light like we do other broadcast pollutants. For example, if our neighbor were blasting their sound system at us all night, the police would be quite willing to cite them for disturbing the peace were someone to complain. But make the same complaint about a light you can't escape and the police might very well arrest you for annoying them. Perhaps the dichotomy is due to the fact that powerful, long lasting, automatic, artificial lights are culturally a very new thing; younger than automobiles. It was not so long ago that, away from a center of population, about the only artificial light one saw at any distance was flickering of candles or fireplaces. Like virtually every other American, I'm pretty addicted to having electricity available, but unlike most of us, about the brightest thing in the room as I write this is my PowerBook's screen. It's currently running with only a single bar of its brightness indicator lit. If we have an energy crisis for routine things, we could easily ameliorate a great deal of it with little effort and expense simply by taking control of power switches on lights and using them only when you specifically want to. As a side note, I recall thinking how nuts it was in movies that people would arrive at their house, walk through the front door and begin a ritual of turning on every light in the house. I thought no one would really do that. But I was wrong. I've been with people who've done exactly that for no apparent reason.

Are you prepared to Clang your LLVM? You most likely won't have to worry about it, but Apple is hard at work doing just that. In the not too distant future, **it's likely that all the new software for Apple's systems**

will be Clang'd via LLVM.

So what? Well, like I said, you most likely won't have to worry about it. But it's something you should understand a little about since it revolves around a rather fundamental aspect of Open Source Software and OSS has a huge effect on the programs available to operate your computer. For a number of years now, most systems have been developed using GNU licensed compilers to develop both operating systems and the software that runs on them. Not too long ago, the GNU project's leader, Richard Stallman, created a new GNU license that makes the production of proprietary systems somewhat problematic. You see, Richard is of the opinion that OSS should not be creating proprietary systems; something that the old license was OK for. So, new versions of GCC (GNU Compiler Collection) can't legally be used the way system vendors have become accustomed to. Solution: start an OSS project to develop a compiler under a license doesn't rock the boat. The approach taken by this project is to first create a processor independent model of a machine that high level languages can be converted to and then write the programs that do those conversions. The result is LLVM (Low Level Virtual Machine). One other step is needed: for each actual processor type, a program has to translate the LLVM code into machine code. Clang is the C-language compiler that analyses the gibberish probably written by some human (C is used by some programming systems as an intermediate language) and translates it into LLVM. In what could be a very interesting twist, there's talk of Apple using a JIT (Just In Time) system that won't pre-translate the LLVM code to machine code. Instead, it's believed that LLVM translation will be so quick, there'll be no need to create many actual "binaries." I find all this somewhat amusing and exasperating at the same time. Twenty years ago, we had systems capable doing this sort of thing.

6

7 -->

That Bane of the Macintosh User

Writing web pages—HTML (HyperText Markup Language) — is a process that is supposed



to follow very well defined rules. Depending on one's perspective, the fact that Web browsers compensate for numerous constructs that do not follow the standards (not a reference to Microsoft's efforts to subvert public standards with their proprietary set of constructs) can be problematic. This is because it can be difficult to test your work against a reliable standard by using any particular browser. Unfortunately, not many web site developers care all that much about writing correct HTML as a result.

But, if you are developing web pages and want to do it right, there are a couple of things I can recommend. One is to use **iCab** as a test browser. There is an error reporting window in **iCab** that can tell you all about how the page you're viewing fails to meet the public, open standard, even though **iCab** may well render the page as well as another browser.

The other thing is something I recently bumped into while trying to find a simple tool to help someone see where they were leaving things out of their HTML. All I really wanted was a "pretty print" — a canonical form of structuring the text so that one has a good chance to grok what's related to what. Up popped **tidy** in a Google search. I'd heard of **tidy** before but never needed it for myself ... I'm disciplined by long practice about writing code so that it's "pretty" enough for me.

7 I'd always heard of **tidy** being a typical GUI application. True enough, you can get a GUI

version of **tidy**, but this is not my preference for such things. Why not? Well, having worked with HTML for a while, I know that I've often needed to apply text transformations over whole folder hierarchies of stuff. This is not something that's a lot of fun to do by mucking about with a mouse: it takes a lot of time, is likely to give a bad case of mouse wrist and, worst of all, will most likely not be done correctly on every file. Why not? One thing a lot practice writing code has taught me is that humans (more specifically, this one, anyway) are not disciplined enough, as a rule, to do large numbers of repetitive tasks without making mistakes. Computers, on the other hand, have a marvelous ability to do exactly what they're programmed to do, over and over and over. So, when I saw that **tidy**'s nominal form for Mac OS X was a "Terminal application," I couldn't pass up investigating it. You can find everything via <http://tidy.sourceforge.net/>.

One curious thing about the Mac OS X distribution of **tidy** (haven't looked at others) is that there's no "man page" included. If you need to read one, there's a web based version at http://tidy.sourceforge.net/docs/tidy_man.html. As near as I can tell without actually doing it all, you can create your own local man page for **tidy** by checking out the code from the CVS archive and building it. Having a "man page" may not be important since you can get a lot of help by typing **tidy** followed by **-help**, **-h** or **-?** to list the command line options. You probably won't need to know about all the stuff **tidy** can do, anyway.

To convert a file you have into a "pretty print" version to look at, you can do something like

```
tidy -iu myfile.html>pretty
```

The **-iu** tells **tidy** to indent (this is the "pretty" part) and uppercase all the tags from **myfile.html** with output to "stdout." Generally, programs read their input from "stdin" (which **tidy**

assigns to **myfile.html** here) output their results to "stdout" and output error messages to "stderr." Since both "stdout" and "stderr" are pointed to the Terminal window by default, the **>** is used to redirect "stdout" to the file **pretty**.

One of the things you'll notice, if there are errors in the code, is copious output of error message text. This can be annoying if you're processing a number of files in an automatic procedure. For example, I like to use something like this to process files in a folder:

```
ls *.html | while read x; do  
tidy -ium "$x" 2>/dev/null;  
done
```

Here, the **ls *.html** command creates a list of names of HTML files in the current working directory (the folder you're working on in Terminal. These names are "piped" (via the **|**) to the **while** loop. The **while** loop is controlled by reading file names from the pipe until they run out (ie. all the HTML files have been processed). The file names are read into the variable **x**. This particular **tidy** command pretty prints and uppercases tags in the files and puts the result back into the source file. I normally recommend against doing this latter function. It should only be done if you have a good backup of the files you start with. The **2>/dev/null** part redirects error output from the **tidy** into the big bottomless bit bucket so you don't have to be confused by all the messages about everything you've messed up in the original files. **8 -->**

<--Water Blogged Wump

They just weren't in the mainstream of geekdom. It's a curious thing we humans do - it matters not at all how good something is; only how popular. LLVM is an overtly complex mess from my perspective. It's clearly been defined by the belief that supporting the current melange of bad ideas is a requirement.



<-1 Keeping In Touch

the two of us! Wow! We chatted for perhaps 30 minutes. This with no long distance charges! My friend in Phoenix and I chatted about once a week using a dial-up connection. I finally got a faster modem and things worked much better.

Now, once in a while I use an application called **Adium** to accomplish the same thing I did back in the 1990s with a dial-up connection. The cable connection I now have is much better, but a dial-up would work too, if that was all I had. Other applications will work like **Adium**, but a few of us like

it, so we use it [**Ed: it handles more "chat" protocols than iChat**].

One feature I believe all similar applications have is that you can open the program, indicate you are "active" and still do other things on your computer at the same time. If someone else, that you "accept" sees that you are "active" they will let you know via the software. Then you can have a great text chat. In some cases you can even have a vocal chat. My internet connection doesn't let me have vocal **iChats**, but I don't mind using text. Text leaves a record of what was said and is thus helpful later to go back and get some

information someone gave you that you forgot about. Yeah, I am getting older and the memory isn't what it used to be!

How about you getting on-line with an active button clicked, so the rest of us can join you in a chat? This is an excellent way to get help with a computer problem, to send files too big for email, or to just share ideas. It's easy and free. If you already have been using on-line chatting, log on and join us!

If you have questions, please email either me at mcclellan@charter.net or Walt at walt@wump.org.

<--Dread — tidy, pg 7

Here's a simple example of what **tidy** can do for you. I created a very badly formed HTML file with the following:

```
mobi:~ wump$ cat > x.html
<html>
<head>
<title>
Dumb Test
<body>
What?
```

Then ran **tidy** and got the output on the right. The error message text was output first and then the text of the transformed html file. As you can see, tag pairs have been completed and standard informational tags have been added. The tag type words have all been upper-cased and the layout of the text is "pretty."

```
mobi:~ wump$ tidy -iu x.html
line 1 column 1 - Warning: missing <!DOCTYPE> declaration
line 3 column 1 - Warning: missing </title> before <body>
Info: Document content looks like HTML 3.2
2 warnings, 0 errors were found!
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

```
<HTML>
<HEAD>
  <META name="generator" content=
  "HTML Tidy for Mac OS X (vers 1st July 2004), see www.
  w3.org">
```

```
<TITLE>Dumb Test</TITLE>
</HEAD>
```

```
<BODY>
  What?
</BODY>
</HTML>
```

Coming Soon to an ABCC Near You!

We still have one more meeting date planned ahead. September 18, Ian Schray of Softpress Systems, Ltd., will present Freeway 5.1: software for building websites.

As amazing as that is, it's incumbent upon us to dream up what we want to meet for in the future. It's therefore suggested that if you'd like to have something coming soon to an ABCC near you, that you come

up with at least some ideas as to what you like to have come your way.

Sure, you'd like a seminar on guaranteed methods of picking lotto winners using your Mac. Or, perhaps something on how to make a fortune with your computer working one evening per week. Great ideas but perhaps we might actually be able to get some programs if the ideas happened to be practical. Dream on. Take notes.

unClassifieds

Need a manual?

Discount for ABCC members!



Apple Blossom Computer Club

<http://www.abccmug.org>

Give it a look.

Put in your own...



ABCC near you, that you come