C·O·N·T·E·N·T·S

GENERAL MEETING OF AUGUST 25, 1990

MAIN PRESENTATION
"The Birth Of Silicon Valley"

Thanks to Jan Altman we were fortunate to have two of the leading participants from the 'early days' of the personal computing scene.

JIM WARREN

He kept the meeting enthralled as he led us down memory lane in a series of Connections involving the interactions between the pioneers in the Valley. The trip began at Stanford where, (after his famous nude parties became rather more than the Catholic college, where he taught math, could tolerate), he studied computing. Here the Midpeninsular Free University was formed which was a focal point for the forthcoming computer explosion. The treasurer was Larry Tessler who went on to Xerox PARC and worked on the WIMP environment which Apple grabbed for the Lisa etc. At FU he met Bob Albrecht who was involved in the setting up, with Dick Raymond, of The Portola Institute (specializing in funding projects to bring compute power to the people) which in its turn begat the Whole Earth Catalogue. Albrecht went on to found the tabloid Peoples Computer Company, and then, with Dennis Allison, Dr. Dobbs Journal of which Jim Warren was the editor.

He was an early member of the Homebrew Club (which started in March '75) when Lee Felsenstein was moderating the Mapping Session. Where John Draper (Captain Crunch himself) would discuss ideas with Steve Wozniac and Dan Sokol would barter 8080 chips. Where Woz was getting his first look at data sheets for the 6502 [designed by Chuck Peddle]. Where George Morrow and Bill Godbout were looking for opportunities and Adam Osborne was selling his books out of a box.

Jim was one of the 3,500 visitors to John Dilkes' "Personal Computing '76" festival in Atlantic City. This huge attendance for the first event of its kind inspired him, the following year, to start the enormously successful West Coast Computer Fair (the first one was held in the San Francisco Civic Center in April '77). Jim Warren gave us a taste of what it was like to be in at the start of the personal computing revolution when idealism was all and Lee Felsenstein's philosophy that computing was for the people, with information and software freely available, held sway.

CHUCK PEDDLE

He then gave us the view from the commercial developer's angle. He designed the 6502 chip (which Steve Wozniac bought at Wescon where Chuck had rented a hotel room - because you couldn't retail at the show itself). The following year, at Jim Warren's West Coast Computer Fair, Apple showed its Apple II while Chuck exhibited his own design - the Commodore PET.

He worked for a short while at Apple which, while he was working at Commodore, he had tried to persuade Jack Tramiel to buy. Tramiel didn't see much future in a personal computer. Chuck, the entrepreneur (he went on to found - among others - Victor Technologies/NNA [No Name Available]/This Tyme), reminded us that the birth of Silicon Valley was due to the calculator or, more precisely the development of the chips, that made the calculator possible. That Pong, designed by Nolan Bushnell who founded Atari, was the first computer game, made a hugh amount of money, and opened the way for the market driven industry which became the real start of Silicon Valley as the Hi-Tech center.

Chuck ended his talk by making the point that you can get anything in the Valley. For any information on any subject there will always be an available expert on it "walking down the street". That's the marvelous thing about Silicon Valley.

Between them they gave us a taste of what it must have been like in those early days with all that incredible synergy, when all the ideas were fresh, and all possibilities lay ahead. A very entertaining and enlightening evening; Jim Warren and Chuck Peddle pretty much know where all the bodies are buried. ☐2

ADDENDUM

For anyone wishing to find out more, I refer you to Steven Levy's book 'Hackers' Doubleday 1984 (HB) Dell 1985 (PB), probably the best account of what computing was like in the early days (1950's onward). For an account specifically about Silicon Valley see 'Fire In The Valley' by Freiberger & Swaine, Osborne/McGraw-Hill 1984 (PB).

There is also a rather nice genealogy poster which shows the timeline interaction of people and companies in the Valley. It is published by SEMI, of Mountain View. The current version covers changes up to 1986, though there is talk of an update.

The Editor would like to hear from anyone who is interested in any aspects of the history and development of computing. ⌐z

SIG NEWS

Microsoft Word 5 SIG meets on the fourth Monday of each month at 7:30 pm. On Monday, October 22, we will talk about how to merge form letters.

Word for Windows/Windows SIG meets on the second Wednesday of each month at 7:30 pm to talk about Word for Windows and other general Windows issues. On Wednesday, October 10, we will talk about the table feature.

Location of all meetings is Infotec Training Institute, Techmart, 5201 Great America Parkway, Suite 254, Santa Clara. For more information, please call Jan Altman, (408) 243-5955. ⌐z

MEMBERS RESPONSE

Don Baird

Such questions as those below will, hopefully, continue helping to acquaint members with each other. Each month three members are selected at random to answer any three of the following questions.

1) The most intriguing aspect of personal computing to you? 2) Most interesting or useful activities of the club? 3) Prefer what additions or subtractions from club activities? 4) What uses your computer is put to? 5) What should the club emphasize? Social; educational; entertaining; purchase deals; activist, for example: towards better programming and documentation of software; other. 6) Should we try to set up structures to make it easier for members to help each other?

Larry Mehl: For 1), he finds communication the most intriguing; like the sharing of information inside and outside of the club such as with the bulletin board and networks. For 2), the most useful is being face-to-face with people in the industry like our August program where one could go up and talk with the speakers. The reply for 4), indicated mostly for business (database, spreadsheet, and word processing).

Mike Ardley: For 1), is intrigued that the computer can do things that are impossible by other ways. For 4), he mainly uses it for business, experimentation, word processing, and spread sheets. He believes 6) should be the top priority of the club.

Laddie Hughes: On 1), most intriguing is that it is a new technology to her. She likes to keep up with what's going on in the world, and this is one aspect. She suspects that the history heard from the August speakers was the way things were, but it was interesting to hear it from them. On 5), says must emphasize that people are on different learning curves at all times, and that we must try to serve all of them. Believes that this is difficult but necessary. On 6), says that the structures might already be there. Says anytime she asked for help, she has gotten it. Feels that such structures are important. ⌐z

REFLECTIONS

Don Baird

We might be on the threshold of the photonics revolution. ("photonics" by now must be coined in the optic/electronics industry: if not, then I just invented it). While electronic developments blazed along after Dr. Lee DeForest's vacuum tube break-through. and then really picked up speed with Bardeen et al transistor, we are now having rather quiet but important intrusions of optics devices which are supplanting electronics. Each step often is a break-through. The lightning-fast development of the rudimentary and bulky crystal laser into the tiny laser diode already resulted in a computer chip having 1 million lasers per square centimeter. RCA's early approach to television was prematurely fashioned on how the organic eye was thought to work, which burdened their TV system with huge and impracticable bundles of wire cable. Perhaps, this time, lightpipe/laser arrays will permit a much better approximation. Rather than limited to millimeter wavelengths, we will be dealing in Angstroms.

Even though laser developments use techniques laboriously learned while building electronic integrated circuits, it could still take a long time to get through the "photronic" period we are presently in 'til reaching a truly "photonic" age. It's easy to envision plug-in boards (or more likely cubes because of nil heat to dissipate) using lightpipe bundles to interconnect functional units. However, until the breakthroughs delivering a photon flip-flop of micron dimensions, optical computer logic will be primitive. Other breakthroughs for basic functions are required, such as are analogous to resistance, capacitance, and inductance. It will be interesting to see what an optical ROM becomes.

We may never experience total photonics. Just as electronics always used chemistry to supply a storage function (the battery), predominate optical apparatus of the future may always be hybrid.

The perspectives of the foregoing stem much from personal involvement. I can remember clearly, sitting on the porch with an article from Popular Mechanics diagraming how to receive radio signals from using a jack-knife blade/tickler wire as the detector, a long wire hung from a tree, and a pair of earphones (now called head-sets). The nearest station was KSTP in Superior, Wisconsin, 72 miles to the West. Learning much later that a razor blade was better, still produced no results. Still later, I found that my Dad's radio course came with a piece of Galena (lead-sulphate crystal). My first real thrill in electronics came when the first faint signals were heard in the earphones while delicately adjusting the tickler wire ("cat's whisker") on the surface of the Galena.

These were early days - before the Great Depression. My town had no radio station. I progressed rapidly by adding tuning to the circuits allowing me to pick up WGN (Chicago) and WCCO (Minneapolis). Reading my Dad's course quickly promoted me to using a vacuum tube (a Radiotron O1A). Pretty soon, I was experimenting with super-regeneration, and sometimes felt that I was scooping the radio world, especially when I discovered that I could dispense with a B battery (plate voltage supply) and use only an A battery (filament supply). I wound short-wave coils and the seemingly vast and mysterious world of short-wave was entered.

At that time, I didn't know what a transmitter was. One day (I was seven), the radio downstairs let out a whistle while I was adjusting the super-regeneration on my one-tube receiver. I still didn't know, but began

to suspect that this might be what radio transmission was all about. In minutes, I learned that by tuning the signal away from other broadcast stations, just a carrier could be heard. In breathless excitement, I put one of the earphones in series with the grid circuit and spoke into it giving me the second big thrill in radio when my voice come from the radio downstairs.

I was doomed. I haunted radio repair shops and was given discarded radios (I can't bear the thought of what those sets would now bring on the antique market). Many years later, I counseled myself not to make a hobby also a profession - but to no avail. As part of it, I've closely tracked the electronic age in as many ways as anyone could. My resume became implausible - until eventual full-time consulting required only a client list.

So now, each cycle is similar but unique. Each has its subcycles: Electronics with its developments in vacuum tubes (hot and cold cathode, multipurpose, subminiature, noise sources, tuning eyes, phase splitting, CRTs, TV cameras, cascode, photo multipliers, magnetrons, klystrons, traveling wave tubes, backward and forward wave oscillators); and even in the simple diode (linear, step, low noise, square-law, power, high voltage, vacuum, silicon, germanium, gallium arsenide, colored light, and laser). Solid state discoveries first brought the transistor which then went through transformations of such magnitude that it would be foolish to enumerate them here. The so-called discrete circuits these devices were used in often took a variety of forms that stretched belief. Many are now embodied as integrated circuits. Talk of going molecular seems pretty far out, but I felt the same when observing early attempts at integrating the circuitry.

The key discovery that made the electronics industry possible was the vacuum tube amplifier. The British more accurately called it a valve. Neither was strictly true, because the phenomenon really was the analog control of a high current flow by a lesser current flow. I just now was pondering whether it was possible to do likewise for the photonic age - then remembered that I did it myself photronically about ten years ago by the successful design for sub-nanosecond and precision analog light output control of a power laser diode. As with other like circumstances, the import of what I had done was lost amid all the other designs I cranked out.

Once key functions are cracked, products proceed from large, inefficient, and costly towards small, efficient, and cheap. The effort, time, and billions spent in between is a panorama not always easy to track until it's almost over. It's not likely that anyone will be able to anticipate reliably from here on, because computers are now spewing out unexpected answers. Even a decade ago, computer programs were solving algebraic equations that mathematicians could not. One branch of artificial intelligence has already found that simulating evolution by constructing a genetic code model which then gets altered by pseudorandom selection has given optimum results in less than 100 genetic changes compared to trillions of trials required when using conventional optimization methods.

Unfortunately (and I hope I'm wrong), the sheer esoterism and complexity of technologies may render it almost impossible for some young person to again get thrilled over something as simple as a piece of wire and a knife blade.

RIGHT ON THE MARK

or:

I Formatted a Paragraph and Survived

Jan Altman

Don't Tell Me, I Already Know

Very strange and bizarre things happen when you try to format a paragraph. You turn on a paragraph format and then can't turn it off. Your brand-new paragraph shows up with all sorts of formatting that you never gave it. You merely press the *backspace* key and your document appears to enter the Twilight Zone.

Well, believe it or not, paragraph formatting in Word is really, truly user-friendly. That is, as long as you understand a few Golden Rules that Word uses. These rules are not only simple to learn, but once you adopt them, I promise that Rod Serling's voice will never haunt you again.

Defining a Paragraph

The term "paragraph" may not necessarily mean the same thing to Word as it does to you and me. In Word, a paragraph is determined by the placement of paragraph marks. (You know - the little symbols that look like Roman columns: "¶". They're inserted into your text whenever you press the Enter key.) In order to see the paragraph marks on your screen, pull up the Options menu, and under "show non-printing symbols," choose either Partial or All.

A paragraph is defined as everything between one mark and the next, *includ-*ing the ending mark. When you press the Enter key, you insert a paragraph mark and end a paragraph. If you've pressed the Enter key several times to space down in your document, each one is a separate, single paragraph. A somewhat picky but extremely key piece of information in understanding paragraph formatting.

Positioning the Highlight

Applying a format to a paragraph is one of the easiest things you can do in Word. In just one step, you can format anywhere from one paragraph to all the paragraphs in your document. It all depends upon how you've positioned the highlight.

The First Golden Rule to remember is to *touch the paragraph you want to format*. That is, position the highlight anywhere within it: on the first character, in the middle, or even on the mark at the end. It doesn't matter how many characters are highlighted. As long as you somehow touch the paragraph when you apply the format, the formatting information will be applied correctly. (Remember that this concept is key to *paragraph* formatting. *Character* formatting is handled differently.)

Often, you may need to format more than one paragraph at the same time. In this case, simply make sure that the highlight is somehow touching each one. (An easy way is to hold down Shift while you use the arrow keys to extend the highlight over each paragraph.) Remember that it doesn't matter how much you're highlighting within

each paragraph. As long as you're *touching* each one, it will receive any paragraph formatting you apply. (Want to make your entire document double-spaced? Press Shift/F10 to touch every paragraph, and then apply the double-spaced format.)

APPLYING A PARAGRAPH FORMAT

There are two ways of applying a paragraph format. The standard way is via a menu. Paragraph formats can be found through five different Word menus:

Format Paragraph

Format Tab

Format Border

Format Running-Head

Format pOsition

These menus all apply paragraph formats. This is important, because when you use a command from these menus, Word knows to format the entire paragraph(s) that you're touching. Once you've selected the text you want to format, simply pull up the appropriate menu, make your choices, and press Enter.

Many paragraph formats can also be applied by using the speed formatting keys. The speed formatting keys are shortcuts to some of the more common formats found on the menus. To use one, simply hold down the Alt key and press a one- or two-character code. (Speed formatting keys are an integral part of my work with Word. They not only save me mega-keystrokes, but they also make me look like a sophisticated

power-user to anyone who's looking over my shoulder.) You can find a list of the speed formatting keys on your Word template, or in the table at the end of this article.

A STOREHOUSE OF INFORMATION

Once you've formatted a paragraph, the Second Golden Rule to remember is that *all of the formatting information is contained within the paragraph mark*. That is, that little Roman column-type mark on your screen serves as a sort of storehouse for all of the formats you've applied to that particular paragraph. Want to totally demolish the formatting of a paragraph? Simply delete the paragraph mark. With it, you've deleted all the paragraph formats you've applied. (Hence the Twilight Zone effect.)

Remember that if you've accidentally deleted a paragraph mark, recovery is simple with **Insert** (if you used **Delete** and the mark is in the scrap) or **Undo** (if deleting was your very last editing action). But if (horror of horrors) the mark's not in the scrap and deleting was not your last action, I'm afraid you'll have to reformat the paragraph again from the beginning.

CONTROLLING PARAGRAPH FORMATTING

If you press Enter while touching a paragraph, you create a duplicate paragraph mark. That is, the mark you create has the identical formatting instructions as the paragraph you were touching when you created it. If you're sitting at the end of a centered paragraph and press Enter, the new para-

graph following will be centered also. If you're on the first character of a paragraph that's double-spaced and you press the Enter key, you've created a new paragraph above it with the same double-spacing information.

This no doubt accounts for some strange (but very common) occurrences: new paragraphs showing up with unintended formatting, and being unable to turn off a format. From now on when this happens, you'll calmly nod your head, and realize that you simply carried on the formatting of the original paragraph.

Ridding a paragraph of paragraph formatting is simple. You can either:

1) pull up the appropriate menus and turn everything off (long, tedious way), or

2) apply the speed formatting key for a normal (unformatted) paragraph: Alt/ P (much more efficient way).

(Of course, method #1 is desired if you need to turn off only *some* of the existing formats, but leave others in place.)

PLANNING MEETING

Oct 8 (2nd Monday), 7:30pm
'Phone **Beverly Altman,**
(415) 329-8252, or **Don Baird**
(415) 365-6822 for the location
Help us make decisions. All
members are welcome, you
don't have to be a club officer to
get your views heard.

FOR ADVANCED FORMATTERS:
COPYING FORMATTING AND OTHER TIPS

* Once you've formatted a paragraph, you can easily apply the same format to other paragraphs by using F4, the repeat key. The process is very simple:

1) Apply a format to the first paragraph (either via menus or speed formatting keys).

2) Select the next paragraph to be formatted (**F10** and **Ctrl/End** jump quickly by paragraphs).

3) Press **F4** to repeat the change.

* The above technique is fine if formatting the original paragraph is your most recent editing change. If it's not, there's still a slick way to copy the formatting information from *any* existing paragraph to any other:

1) Position the highlight on the paragraph *mark* that contains the desired formatting information.

2) Press **Alt/F3** to copy the mark into the scrap (pertinent paragraph formatting info is now captured in the scrap).

3) Position the highlight on the mark of the paragraph to receive the new formatting.

4) Press **Shift/Insert** to replace the selected paragraph mark with the one in the scrap. The current paragraph, being given a new mark, is totally reformatted.

* Don't forget to use the new-line character (**Shift/Enter**) at the end of each line when creating tables. The new-

line character creates a new line *without* creating a new paragraph. (This makes paragraph formatting much easier.)

* Mouse users have an extra-added advantage. To copy paragraph formatting using a mouse, merely do the following:

1) Touch the paragraph(s) you want to format.

2) Using the mouse pointer, go into the *selection bar* (that's the skinny area between the left window border and the left edge of your text where your pointer becomes a right-handed arrow) alongside the paragraph that has the format you want to copy.

3) Hold down Alt and click the right mouse button.

4) Voila.

Summary

In summary, paragraph formatting is fast and efficient if you remember these few simple rules:

1) Word determines a paragraph by the placement of paragraph marks.

2) The highlight must merely touch the paragraph or paragraphs you want to format.

3) All of the formatting information is contained within the paragraph mark. (Pressing Enter while touching a paragraph creates a duplicate mark.)　　　　　　□z

WORD 5 SPEED FORMATTING KEYS	
ALT/L	Flush left
ALT/R	Flush right
ALT/C	Center
ALT/J	Justify
ALT/N	Increase left indent
ALT/M	Reduce left indent
ALT/Q	Increase left and right indent
ALT/F	Indent first line
ALT/T	Hanging indent
ALT/2	Double space
ALT/O	Blank line above paragraph
ALT/P	Normal (unformatted) paragraph
ALT/F1	Set tabs
CTRL/F2	Create running head
ALT/F2	Create running foot

NOTE: When using a style sheet, you must insert an "X" between the ALT key and the code.

Jan Altman is a PC trainer and consultant who specializes in Microsoft products, and leads special interest groups in Word 5 and Windows applications. She also edits books on software. Jan can be reached at The Express Train, 3655 Pruneridge Avenue, No. 135, Santa Clara, 95051, or at 408/243-5955.

TECHNICAL TOPICS

WORKING IN THE ENVIRONMENT

Tony Allen

For those users unfamiliar with the uses and purpose of environment strings, they are named text strings stored in memory by DOS, which can be accessed by DOS commands and other programs. Examples of environment strings are COMSPEC (used by DOS to determine where COMMAND.COM can be loaded from), PATH (the directory search path used by DOS to find executable programs) and PROMPT (the user-defined prompt format). Some environment strings are created by their own special commands (such as PATH and PROMPT), or by the DOS SET command.

A "master" copy of the current environment strings is stored in your PC's memory. A copy of it is made, and placed in high memory, for every program executed. The program can then access its own copies of these strings, and determine their values. For example, a program might use an environment string call "TEMP", which would be defined by the SET command to be the drive\directory path where temporary files might be stored. For example:

```
SET TEMP=C:\PM\Pagemakr.
```

Many software programs use environment strings to store information such as terminal types; most compilers and program development tools use environment strings to tell them where to find libraries, source code, and other items. A list of the current environment strings can be displayed at the DOS prompt by entering the command SET, without parameters.

INCREASING ENVIRONMENT SPACE

If you think of DOS (an acronym for Disk Operating System) as the *kernel* of the computer operating system, then the *shell* surrounding it is the program COMMAND.COM. It is this that you interact with when you communicate with your system, not DOS itself.

COMMAND.COM creates a set of environment strings within itself, based on entries in CONFIG.SYS and AUTOEXEC.BAT. The maximum size of this environment is 32K and the minimum (the DOS default) is 160 bytes. Now 160 bytes is way too small for a decent path etc., and 32K is a waste of memory (remember all this is 'below the 640K line' memory) so I suggest 512 bytes as a good size. It will cover most normal uses, and you can always enlarge it if the dreaded "out of environment space" message flashes up. Now to change the size of the environment to 512 bytes put the following line in your CONFIG.SYS:

```
SHELL= COMMAND.COM /E:512 /P
/E stands for Environment
/P makes the shell permanent
```

NOTE:

If you are using DOS 3.1:

a) you should upgrade;

b) you will need to specify the size in multiples of 16 (that good old Hex number) so a 512 byte environment would be 512/16 = 32 and the line would read:

```
SHELL= COMMAND.COM /E:32 /P
```

Now, after rebooting, typing SET on the command line will produce something like this:

```
COMSPEC=C:\COMMAND.COM
PATH=C:\UTIL;C:\DOS;C:\PM;D:\QC;C:\;D:\;
PROMPT=$p $d $t$h$h$h$h$h$h_
SET TEMP=C:\PM\Pagemakr
INCLUDE=D:\QC\LIB
```

SAVING MEMORY

One point about the environment and memory allocation, is that you can save a few bytes of memory if you juggle the makeup of your AUTOEXEC.BAT file. By loading all your memory resident programs *before* your Path, or any features that involve the environment, you will save a few bytes of memory. The reason is that whenever a program is run it makes a copy of the current environment for its own use. So by loading all your TSRs before putting anything in the environment space you save that memory. Check the free memory you have now after booting, then amend your AUTOEXEC.BAT, and see just what you have saved. The total amount will depend on the number of programs you load above the path etc.   ⊡z

DID YOU KNOW?

That **DIR .bat** is the equivalent of **DIR *.bat** and that **DIR x*.** is the same as **DIR x*.*.** There you are, two keystrokes saved already, and shifted ones at that.

Next time - the voice operated computer!

TRICKS & TIPS

TURNING IT OFF   *Tony Allen*

If you ever want to blank the screen for any reason this is a small batch file that will do it.

```
copy con OFF.BAT
@echo off
CLS
PAUSE >NUL <F6>
```

The trick here is that the PAUSE command displays the message 'Strike a key when ready...' but >NUL send this message into the black hole of the NULL device. So the screen is blank and it doesn't matter which key you hit to get it back.   ⊡z

KEEP OUT!   *Tony Allen*

SUBDIRECTORIES

The IBM Extended Character Set extends the ASCII character set (0-127) to 255. Character 255 is a null character and can be accessed by holding down the Alt key while tapping out 255 on the numeric keyboard. It produces a blank space.

So you can create a directory by MD\\<Alt-255> and then the command CD\\<Alt-255> will show C:\\ > only on the screen (if your PROMPT string has '$p' in it to show the current directory). Of course, running a DIR of the root directory will show that there is a directory there:

WW	<DIR>	8-13-90	6:48p
BAT	<DIR>	8-13-90	5:36p
DOS	<DIR>	8-14-90	5:53p
TMP	<DIR>	8-28-90	12:20p
	<DIR>	9-07-90	5:08p

but if you don't how to access it you can't get in. It works just the same as any other directory with the exception that some programs (Microsoft Word is one) will not recognize its existence and will not retrieve files from it.

FILES

Reverting to the standard ASCII character set (characters 0-127), the symbol Δ (Alt-127) is a character that is not typeable from the command line. It is a backspace command and therefore deletes itself whenever it is typed - neat huh? OK so how to use it in a file name or a directory name? The trick is to use a word processor/text editor that will generate the Alt-127 character (Microsoft Word will do it, as will Edlin - Sidekick will not). The following two batch files - created in Word do the job of:

a) converting the file name to an inaccessible one and,

b) converting it back (when access to it is needed).

Of course there are a number of ways this trick can be overcome, but for the casual browser it is sufficient to keep them out.

The first batch file RENames the file so it starts with the Alt-127 character. Call it say:

```
HIDEFILE.BAT
@echo off
ren %1 Δ%1
```

Now typing HIDEFILE <filename> will append the Alt-127 character to the first letter of the file.

The second one call:

```
UNHIDE.BAT
@echo off
ren Δ%1 %1
```

This changes the file back to its original name thus allowing you, and your programs, access to it.

Don't forget that while these names make them easy for you to remember, they also make them easy to find for someone else. So perhaps a pair of rather more innocuous names would be better.

SUBDIRECTORIES AGAIN

Making directories inaccessible works in a similar fashion. Unfortunately DOS does not provide a program for renaming directories so these batch files use the PD program RENDIR.COM written by John Dickinson [PC MAGAZINE] which must be accessible (usually via your path).

These three batch files are self explanatory. The caveat on naming them remains the same.

```
HIDIR.BAT
@echo off
rendir %1 Δ%1

GOTODIR.BAT
@echo off
cd Δ%1

UNHIDIR.BAT
@echo off
rendir Δ%1 %1
```
`^z`

QUICK PRINT   *Tony Allen*

Two quick ways to print a short note or an address.

1. Send the keyboard input directly to the printer (rather like turning the computer into a typewriter):

```
copy con prn
The Stanford/Palo Alto PC Users' Group
P.O. Box 3738
Stanford CA 94309
<F6>
```

2. Use copy con to create a file - then send it to the printer:

```
copy con SPAUG
MEMO to the Editor:
The Newsletter needs more tips like this!
<F6>
```

This creates the file SPAUG which you send to the printer with the command:

```
copy SPAUG prn <Return>
```

Neither method is suitable for writing the great novel but very useful when you need to punch out a quick few lines, and don't want to crank up your word processor.

NOTE: The command -

```
copy con <destination>
```

tells the computer to send what is keyed into the CONsole to the named destination. This destination can be a file or a device. In the two above examples the first one sends the address directly to the printer (PRN) while the second sends it to a file (SPAUG). Files, by the way, do not have to have an extension.
`^z`

PROCOMM TIP   *Don Baird*

The PROCOMM phone directory has a limit of 100 entries. Because this directory (PROCOMM.DIR) is portable to other computers using PROCOMM, the listing can be increased for any one computer by putting different PROCOMM.DIR's into respective subdirectories and then copying the one desired (use COPY or XCOPY, not BACKUP) to where PROCOMM.EXE resides.

Caution: the "dialing command" (line 2 on Modem Setup) also gets ported. When copying a directory from someone else, their dialing command will overwrite yours. Retain a record of your original command to re-institute if necessary.

STATIC TIP   *Don Baird*

Don't be a zap. Touch some grounded metal before using your keyboard or before handling plug-in boards, especially after walking across carpets on a dry day.
`^z`

THE SPAUG RESOURCE CENTER

OFFICERS
President	Don Baird	(415) 365-6822
Vice President	Jan Altman	(408) 243-5955
ASSU Representative	Tim Short	(415) 854-2732

MANAGERS
Bulletin Board (SPARC)	Bob Bottini	(415) 369-2086
Financial Manager	Beverly Altman	(415) 329-8252
Librarian - Public Domain Software	Les Weil	(415) 321-5541
Newsletter Editor	Tony Allen	(408) 739-2953

SOFTWARE
Accounting	Larry Mehl	(415) 329-6037
Foxbase	Marie Hooper	(415) 325-1206
Framework	Jim Caldwell	(415) 692-7181
	Richard Orser	(408) 425-0474
Microsoft Products	Jan Altman	(408) 243-5955
R:Base	Larry Mehl	(415) 326-6037
Lotus 1-2-3	Larry Mehl	(415) 326-6037
Symphony	Sally See	(415) 941-1378

LANGUAGES
C	Corwin Nichols	(415) 494-8640
	John Watson	(415) 325-7632
Fortran	John Watson	(415) 325-7632
Pascal	John Watson	(415) 325-7632
Smalltalk	John Watson	(415) 325-7632
Quickbasic	Don Baird	(415) 365-6822

HARDWARE
Epson Printers	Jim Caldwell	(415) 692-7181

[Change of # since last issue]

CLUB CALENDAR - October 1990

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

October 3   First Wednesday - Novice SIG
7:30 PM   Nancy Helmy, (415) 326-1911

October 8   Second Monday - Planning Meeting
7:30 PM   Beverly Altman, (415) 329-8252
                Don Baird, (415) 365-6822

October 10   Second Wednesday - Word for Windows/Windows SIG
7:30 PM   "Creating Tables"
                TechMart, 5201 Great America Parkway
                Suite 254, Santa Clara
                Jan Altman, (408) 243-5955

October 18   Third Thursday* - Investment Software SIG
7:30 PM   Nancy Helmy, (415) 326-1911
                Bob Phillips, (415) 941-3733

October 22   Fourth Monday - Word 5 SIG
7:30 PM   "Merging Form Letters"
                TechMart, 5201 Great America Parkway
                Suite 254, Santa Clara
                Jan Altman, (408) 243-5955

October 31   Last Wednesday - GENERAL MEETING
7:30 PM   October's presentation - "PC-TOOLS V6"

Normally meets third Tuesday

The Stanford/Palo Alto
PC Users' Group

P.O. Box 3738,
Stanford, CA 94309