

# PolyLetter



PolyLetter 91/1

Page 1

JAN/FEB 1990

## *Editorial*

As you know, our ranks have been thinning over the years. PolyLetter now regularly publishes tidbits about the PC world as well as items on converting between Poly and PC BASICS. Of course, there are die-hards among us who would never dirty our hands with a PC. I sympathize with both camps. I'd like to continue to use my Poly for years to come; I probably will as long as it keeps running, and, like Jim Salinger, I have a stock of spare parts just for that purpose.

But, I also have a PC clone and do use it as a supplement to the Poly. Of course, I also have Bob Bybee's PolyMorphic Systems Emulator (PM.EXE). But on my particular hardware that runs slower than I'd like, and I am not yet prepared to put the necessary money into a high speed PC hardware upgrade. Having PM.EXE is also like having a back-up Poly system disk. One must constantly update it to prevent it from getting out of date. PM.EXE, it seems to me, has its best use in minimizing the pain of transition between Poly's and PC's. Anyone with a large investment in Poly software could continue to use that software in the hybrid PC-Poly environment while slowly converting operations to PC-DOS a little at a time. Still, there's something to be said for the "purist".

To serve our diminishing readers better, I am devoting a part of this issue to a feed-back form which you can remove, fill out, and mail back to me. There's space to let me know what kind of system you are using, what kinds of things you'd like to hear more of, etc., etc. The best way I can serve your needs is if you tell me what you want in future issues of PolyLetter.

## *Announcements*

Melisco Marketing announces the availability of Pony Express version 1.50

with the new postal rates. Pony Express runs under PC-DOS and can be run as a pop-up TSR or as a stand alone program. It provides instant access to USPS mailing and UPS shipping rates, as well as delivery times. Pony Express is available as "shareware" (try before you buy) from most shareware sources, or it can be ordered direct from Melisco Marketing, (800) 642-5045, on a 60 day money back guarantee basis. The registration price is \$50 (\$80 with a printed manual). Pony Express was rated by PC World as a "Best of shareware" product.

Blazie Engineering, 3660 Mill Green Road, Street, MD, 21154, (301) 879-4944, announces a DOS compatible, battery powered 3-1/2" disk drive which connects to any RS-232 port. \$520 + SH.

Electronic Technologies, 3985 South Rochester Road, Suite H, Rochester, MI 48307, announces BarZIP, an IBM-PC compatible software program that prints postal barcodes on labels and envelopes.

Beginning midyear the Postal Service will allow a 2-cent discount for letter envelopes that include a nine digit (ZIP+4) ZIP code and a preprinted postal barcode. The addition of a preprinted postal barcode can also increase the speed and accuracy of mail delivery.

BarZIP is a small memory-resident program that can add postal barcode printing capability to existing label printing programs, database programs, accounting programs and many other software programs. BarZIP retails for \$195. For more information contact Charles Eglinton (313) 656-0630.

## *Letters*

Ralph,

January 16, 1991

"We are now using Bob Bybee's emulator,

version 2.0, on both 33Mhz 386 and 25Mhz 486 machines. Normal processes which had taken 2.5 hours on the Poly now take 20 min on the Emulator running on this hardware. The only thing that takes just as long as it did before is backing up to floppies." -- Tom Bucy, Burbank CA.

[Tom telephoned PolyLetter. Tom Bucy runs Bucy DICASTING and the Poly is used to serve the companies accounting, payroll, accounts payable, accounts receivable, -- the whole nine yards. Tom has a system with many safeguards which backs up to floppies as it goes.

Tom, you can speed up the process by backing up to virtual Poly backup disks (pc files) and then using the DOS "COPY" command to copy the backup virtual disks to floppies. -- Ed.]

Dear Ralph, January 27, 1991

You asked why I agreed not to compete with PolyMorphic's hard disk products. Here's the story.

When I turned PolyLetter over to Frank Stearns, at the end of 1983, it was partly because I wanted to pursue the business of building Poly add-on products, specifically hard disks. I knew this would put me into competition with PolyMorphic Systems. I felt that PL's editor should be on good terms with PolyMorphic in order to receive information on the state of the company, new products, etc., and I did not believe I would be considered a "friend" once my HD hit the market. After all, I was pricing mine at less than half of theirs. I felt my price was more fair, and yes, this is the essence of competition: it causes efficient use of resources and lowers prices.

In April '86, things had changed. I felt I had more to gain by making peace with Poly, than by driving any more nails into their coffin. So I agreed not to sell any more HD's or RAM cards. In exchange for this, I hoped Poly would share with me (and the rest of our group) two things... a list of known Poly users, and the source code to the Poly software, so it could be saved for posterity and ported to new hardware. Alas, neither of these things came to pass.

I was not attempting to eliminate competition. Actually, at that point,

neither PolyMorphic nor I were selling any hard disks anyway, so it was perhaps a useless gesture. -- Bob Bybee, Stone Mountain, GA.

Dear Ralph February 25, 1991

After all of your patience, suggestions, and help in solving my HD problems, I owe you a letter to capsulize the experience and process. The solution for any Poly user with a defective HD and/or controller board, is to buy a controller board from Omnishore Electronics, 1700 Forrest Way, Carson City, NV 89706-0311, and an ST-225 20 Megabyte HD (Seagate Technology).

The price of the board is \$100 and the drive about \$209. Incidentally, the installation of the HD does not permit using the front baffle from the HD box, but full-height face-plates are available.

Installation requires using the correct HD system disk. My system disk is Exec/96 (11/23/82), same as used with a full height RODIME drive. In Exec mode, I changed parameters of the WCU Syquest option according to the data given in the owner's manual with your public domain Hex conversion program. The changed WCU data does not show up on the WCU menu, but it sends the data to the drive and pops up after booting and is ready for the CONFIGURE and V-SETUP programs. All this seems simple now, but you know the difficulty I had in getting through the maze.

My troubles started suddenly when a full height Quantum HD and the control board failed. At the time, the source of the trouble was still a mystery. I called QUANTUM for help, only to learn that my model is out of production. I also called Sirius Parsaei at about 9 A.M. one Saturday morning and was embarrassed to have awakened him. Either he or Quantum put me onto Omnishore.

Next I made the mistake of hooking that control board to a then functioning TULIN HD. Then when the TULIN was hooked back to what I thought was a "good" control board, it no longer functioned. At this pint I sent the boards and drives to Bob Bybee who couldn't make them work. Next we agreed to buy a new control board from Omnishore, but it didn't work either.

Bob wanted me to switch my operations to a PC using his conversion process for my Poly programs. The problem with his suggestion is that I cannot afford downtime in my business and do not want to depend on outside service people to fix a PC or to go to the expense of buying a PC plus a back-up. As you know, I have four Polys, back-up boards, disk drives, and power supplies to fix any problem that crops up. In twelve years, I have fallen in love and have the utmost confidence in the Poly.

My next step was to buy the ST-225 HD -- only to find that I couldn't configure with the control board which was supposed to work.

Omnishore wanted me to return the board for testing, but I decided to gamble by buying another control board on the theory that the other board was knocked out by the defective TULIN. As you now know, the newest board works fine. I asked Omnishore to test the other board and am awaiting their reply.

If anyone else gets into a similar pickle, I would be glad to discuss the problem hoping to save them three months of struggling like I did. -- Again, many, many thanks and best wishes, -- Jim Salinger, Cincinnati Ohio.

[Jim, I'm glad that your troubles are finally over and that I could be of some help. For our other readers, Omnishore's address was published in PolyLetter 90/5, and for those who may have want it, here is their phone number: Omnishore: (702) 883-8885. Omnishore was once a subsidiary of Xebec and still manufactures and repairs the Xebec hard disk controllers. If you don't yet have a Poly HD, you would also need an S-100 host adapter, in addition to the controller, in order to add a hard disk to your Poly. -- Ed.]

### ***Some Observations***

by Charles Steinhauser

Well I see the PolyLetter ranks have dwindled down to 28 or so subscribers, and I suspect that only double that amount still uses a Poly regularly, for business or otherwise. It is really surprising to see that the Poly has been around this long and still doing useful work; it wasn't always an over-achiever. When first introduced it was, for the most part, state

of the art hardware. But the engineers ran the company totally. This was Poly's first big mistake; engineers typically don't know much about marketing and the like. The next big mistake was the decision to use hard sector disks and, lastly, to adhere to the Poly operating system for so long. They finally made a mod to the CPU so you could run CPM. Big deal, look were CPM is now. Well, they say hindsight is 20/20 -- so it is.

I understand the reason for the declining use of the Poly, and it isn't the unreliability of the hardware or the software. It's the difficulty in upgrading the nice walnut box. First there was the memory upgrade. When you first bought the Poly it came with 16K, and that was expensive. If you had 64K in the machine you had a second mortgage on the house. Now, in a 80386-33 MHz system, you can purchase 1 meg, yes 1024K, of memory for 58 dollars. Then we upgraded the floppies in the Poly, that staggering 90K floppy drive had to go. So I had the idea and the new drives, Ralph re-wrote the BIOS, we rewired the controller, and all of sudden Poly had three floppies with 200K per drive. But alas, this is less than one drive on a PC, 360K per drive. So you say, ok, we will install a fixed disk on that bad boy and then we'll have something. Well 1800 dollars later, yep, you got 18 megs -- not bad for a Poly 8080 running under 2MHz clock speed. On a 80386 system an eighty five meg drive and controller can be had for under 500 dollars.

Now I know what you're thinking, all the prices you quoted for the Poly are up to 15 years ago, and the prices have come down. Well, that is right but for obvious reasons. Namely obsolete hardware is bought at swap meets for a dime on the dollar, and if you don't want to replace the part with the same obsolete type, then you can buy a new high tech piece for about fifty cents on the dollar of the original part. Therefore, trying to keep an obsolete Poly up and running is a chore many of us don't care to do forever.

The new PC world is absolutely amazing when you look at what you can buy at the price. The performance is unheard of for a desk top box and the software base is virtually endless. For the price of the Poly 15 years ago loaded up, and with a printer you can buy a 386 machine running

at 33 mhz, eighteen times faster clock speed; this is a 64 bit internal, 32 bit external processor. Four megs of RAM, 1.4, 1.2 meg floppy, 100 meg fixed disk, two serial and two parallel ports and VGA video. This system is almost the ultimate in desktop computing, and all will fit in the same footprint as the nice walnut box.

Yeah, that was some computer 15 years ago. It cost 5000 dollars, which is a lot, but look what it would do, wow! Your very own computer on your desk, this was unheard of. Just a few years back the computers were nothing more than vacuum tube circuits. And now I got this Poly on my desktop that makes that vacuum tube box look like an abacus.

Oh, you know something? When this 386 system is fifteen years old, do you suppose it will look like a vacuum tube box -- like the Poly does to the technology of fifteen years its future? Well I can't really say, but if I were to guess I would think that it would. And, no, I wouldn't have the 386 at that time -- too obsolete -- but I would have one of my THREE nice walnut boxes we call Poly still running as they are today, upgrades and all.

Why? There are some things that you just can't bear to sell or discard. There are some things you wish you had back, like your first car or your first computer. The Poly was my first computer system way back in 1978. That was four years before the first IBM type PC, and Poly was even built as early as 1975. Indeed the Poly was slightly ahead of it's time.

So, I will use my 386 system until it too becomes obsolete, which is probably within the next 5 years. It will be sold to purchase some new high-tech unit, but I will still have the Polys, there are just some things.....

### ***Converting Functions***

Poly BASIC allows for the creation of user defined functions. This capability is available in most BASIC's but not as extensively as in Poly BASIC. Poly BASIC permits user defined functions to have multiple lines. Let me give examples of both types.

An example of the former would be a

function to compute the length of the hypotenuse of a right triangle. Remember, the Pythagorean theorem,  $C^2 = A^2 + B^2$ ? If we know the length of sides A and B, we can compute the length of side C as follows:  $C = \text{SQRT}(A^2 + B^2)$ . We can make this a function with the DEF FN statement. Let's call the function H (for hypotenuse). The function must be given two values, the lengths of the other two sides. If we were to define such a function in Poly BASIC, it would look thus:

```
DEF FN H(A,B)=SQRT(A^2+B^2)
```

Here's a little program to compute and print the hypotenuse.

```
10 DEF FN H(A,B)=SQRT(A^2+B^2)
20 INPUT "How long is side A?",A
30 IF A<=0 THEN 80
40 INPUT "How long is side B?",B
50 IF B<=0 THEN 80
60 PRINT "The length of the hypotenuse is",FN H(A,B)
70 GOTO 20
80 STOP
```

Function definitions like this work in most BASIC's. But Poly BASIC allows multiple line functions. In PolyLetter 88/3 there was a BASIC subroutine to compute N! (N factorial). N! is the product of the numbers from N down to 1. The subroutine was:

```
100 REM compute N factorial
110 REM N must be defined earlier.
120 F = 1
130 FOR I = 1 TO N
140 F = F * I
150 NEXT I
160 RETURN
```

In order to use this subroutine, we must first set N to the value of the number we want to compute N! for, we must then GOSUB 120, and finally we must "retrieve" the value of N! in variable F. It would be simpler if we could define a function to compute N!, and we could just use it like the single line function. Poly BASIC allows this by the use of a FN END statement. The function would look like this:

```
110 DEF FN F(X)
120 F = 1
130 FOR I = 1 TO X
140 F = F * I
150 NEXT I
```

```
160 RETURN F
170 FN END
```

Notice, a subroutine has a RETURN statement. A function has a RETURN (value) statement. A function always returns a single value. Here is a small program to compute and display N!

```
10 INPUT "What number shall I compute N! for?",N
20 IF N<1 OR N<>INT(N) THEN 50
30 PRINT N,"! is",FN F(N)
40 GOTO 10
50 STOP
110 DEF FN F(X)
120 F = 1
130 FOR I = 1 TO X
140 F = F * I
150 NEXT I
160 RETURN F
170 FN END
```

In both single line and multiple line functions two things are true. The variables inside the parenthesis are called the "arguments" of the function and are the data passed to the function to work on. Also, the function must return some value -- called simply the value returned. When a function is used to compute a value it is said to be "referenced", or "a function call".

Since most BASIC's don't have multi-line functions, we can't convert a Poly BASIC program which does have a multi-line function directly into the other BASIC. We must first convert the multi-line function into a subroutine.

A function call automatically passes the values given to it to the argument variables. A GOSUB does not, so we must do it ourselves. The function returns a value to the caller. A subroutine does not, so we must do that ourselves too.

The first step in the conversion process is to look at the function and see what variables are used for arguments. Each function reference to be converted to a GOSUB must be preceded by statements which assign the values to the arguments. In the program above, the argument of DEF FN F(X) is "X". Each place in the program where the function is referenced, we must have a statement of the form "X= ..." before replacing the function reference with a GOSUB. In the case of the function above we can see that it uses F to return its

result. Line 30 is PRINT N,"! is",FN F(N) We must have X=N before this reference. Notice that the reference is printed, so we can't just put the GOSUB where the FN F(N) is; we must put it before it. Since the first line of the function is 120, we must add the following before Line 30: "X=N" and "GOSUB 120". When GOSUB 120 is executed the value of "F" is set by the subroutine, so we can replace the reference with its value. Line 30 becomes: PRINT N,"! is",F But we need two more lines before line 30.

```
28 X=N
29 GOSUB 120
```

Also, when the function is converted back into a subroutine, the value returned and both the DEF FN F(X) and FN END must be deleted. The converted program looks like this:

```
10 INPUT "What number shall I compute N! for?",N
20 IF N<1 OR N<>INT(N) THEN 50
28 X=N
29 GOSUB 120
30 PRINT N,"! is",F
40 GOTO 10
50 STOP
120 F = 1
130 FOR I = 1 TO X
140 F = F * I
150 NEXT I
160 RETURN
```

Poly Functions can be converted to IBM and other BASICs. Single line functions can usually be converted directly. Multiline functions must first be converted to subroutines with GOSUB's.

### *HyperText*

HyperText is a data storage and access technique which enables one to skip around in the database by subjects and topics. Here's how it works. In a standard technical work there will be a table of contents, the body of the document, and an index. When you are reading the document and come upon something you need to look up, you must turn to either the table of contents or the index, look up the topic (if it's listed), find the page number, and turn to the page. You must also save your place so you can go back to the article you started with. You may have to do the same thing again with the new article. You may even have to continue with two or three such "digressions" before returning to the

main topic.

HyperText is a computer implementation which simplifies the process greatly. Topics which would normally be listed in the index are highlighted in the body of the document. You just put the cursor on the highlighted topic and press return. The computer does the rest. It saves the page you are now looking at and then turns to the page the topic you selected is on. It saves your having to look up the topic in the index. To return to the previous topic you need only press escape. The HyperText manuals I have seen so far also have general table of contents; but one doesn't have to worry about page numbers -- the computer keeps track of those automatically.

I have been experimenting with designing a hypertext manual using the Poly system commands help files. I have now compiled a HyperText manual which has all the Poly Exec command help files; the commands are all cross-linked to any other. This system would be useful to anyone who was running Exec under Bob Bybee's Poly Emulator. Connecting my Poly Commands HyperText manual help engine in the background as a TSR would allow "popping up" help for the Poly System while running Exec under PM. I hope to expand the manual to eventually include the entire Abstract Systems Help system.

### **Reviews**

PolyLetter has just received issue #6 of "Supermicro" (formerly known as "S-100 Journal"). This issue is billed as the "1991 annual reference issue" and is chock full of hardware bus information and a cross reference for manufacturers of all kinds of hardware products -- including S-100 bus products. This issue reproduces the bus pin-out information for 9 major busses, including the S-100, PC, AT, and EISA buses. Unfortunately, the number of manufacturers of S-100 products has been vastly reduced recently. There were only 8 manufacturers who identified themselves as supporting S-100 bus technology and many of those were parts manufacturers. This issue retails for \$25.00 and is available from Supermicro Journal, P.O. Box 50777, Provo, UT 84605-9905.

PolyLetter has purchased a copy of "TECH Help!" from Flambeaux software, 1147 East

Broadway Suite 56, Glendale CA 91205 (818) 500-0044. "TECH Help!" is a DOS program and a technical database for DOS and BIOS for PC's through 486 systems. It can be used as a stand alone program or as a TSR pop-up utility. The database is phenomenal! It combines in one hyper-linked source the technical information about DOS and BIOS that you would have to search through dozens of publications to find. It's like having the IBM Technical Reference Manual, the Microsoft MS-DOS Programmers Reference Manual, Peter Norton's Programmer's Guide to the IBM PC, and many other sources combined along with a robot librarian to look things up for you!

PolyLetter has also purchased the "DOS Help!" manual. "DOS Help!" has all the information in the DOS manual with a number of indices into the data-base. When "DOS Help!" is made memory resident it can be invoked in two ways. CTRL-H brings up the main menu, or the page you last looked at. CTRL-L will try to look up the word or phrase the cursor is on. Suppose you type "MODE" and forgot what parameters to give it. Pressing CTRL-L will bring up "DOS Help!", automatically looking up the help information on the MODE command. The authors of the data-base have a sense of humor too; interspaced within the wealth of technical information are a few witty comments. "TECH Help!" is also liberally spiced with similar fun remarks.

---

### **Advertising**

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Noncommercial ads by subscribers are free.

---

For Sale: Two new Qume-142 drives (DSDD-360K, 1/2 ht) \$45 each, or trade for one 1.4M 3.5" drive. Also, 500 (count them) 4116 DRAM (16K to 64K upgrade chip) free for the asking; Limit 64 per customer, you pay shipping. Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

---

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA

Please describe your Poly System.

What other computers do you have? Please describe.

What do you use your Poly for mostly?

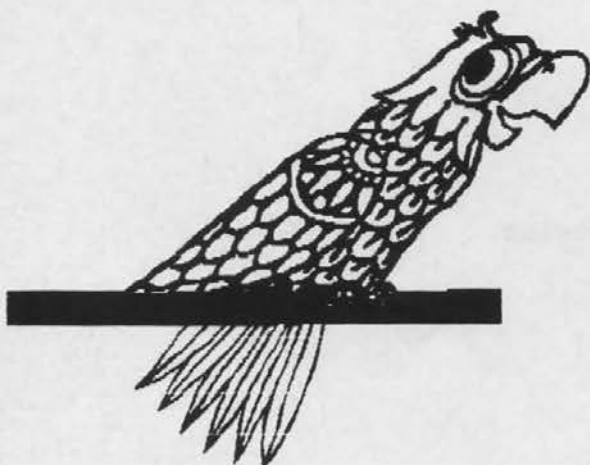
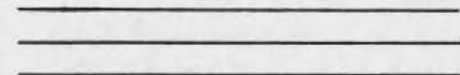
What kinds of Poly articles would you like to see in PolyLetter?

Do you want to see more or fewer DOS related articles? Why?

What kinds of things would you like to have printed in PolyLetter?

Tell us about yourself -- what do you do?

place  
stamp  
here



***PolyLetter***  
191 White Oaks Road  
Williamstown, MA 01267

---

Fold Back here ^ to mail back.

Dear Poly Person,

Please take a moment fill out the survey form on the other side of this page. Any suggestions or feedback you have will be valuable in helping PolyLetter to serve your current and future computing needs.

Feel free to add any additional comments here:



93016-0385

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all) (or swap for something interesting), Doug Schirripa, 716-724-5023 (days) or 716-657-7437. (evening).

Entire PolyMorphic System User Manual, System 88 User's Manual with Exec/96 addendum, & System 88 Operation Essentials On IBM disk. Al Levy, 516-293-8358

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

#### DISKS - MODEMS - PROMS - SOFTWARE - SPELL

1. MAXELL diskettes: 5-1/4" 10 hard sector -- \$10 per box.
2. Used diskettes: 5-1/4" 10 hard sector -- \$0.50 each.
3. Hayes Micromodem 100 (300 baud S-100 internal modem) \$20.  
(If you don't have a modem this is a cheap way to go.)
4. HayesSys modem software (for the Micromodem 100) \$10.
5. Abstract Systems Exec (Enhancements & bugs corrected) \$30.
6. Abstract Systems Proms (Enhancements & bugs corrected) \$35.
7. PolyGlot Library Volumes: \$6 each; 5 or more - \$5 each.
8. Hayes Smartmodem 1200B (IBM compatible internal) \$40.  
Abstract Systems, etc., 191 White Oaks Road,  
Williamstown, MA 01267, Phone: (413) 458-3597  
(Send \$1.00 for a complete catalog--(free with any order).)  
(Make check or money order payable to Ralph Kenyon.)

### HELP!

In this section I share with you the help system files I have built up over the last few years. (The entire system is included with Abstract Systems Exec.)

\$HELP COMMAND fold  
HELP file for system command "fold"

The "fold" command sets upper case keyboard processing.

"fold" sets up the keyboard processor to convert lower case characters to upper case. (see also: "FULL" and "flip")  
"fold" is cancelled by "FULL"

Syntax: "fold"

Minimum size: "fo"  
\$HELP COMMAND flip  
HELP file for system command "flip"

The "flip" command reverses normal keyboard processing.

"flip" sets up the keyboard processor to reverse upper and lower case. (see also: "FULL" and "fold")  
"flip" is cancelled by "FULL".

Syntax: "flip"

Minimum size: "fl"  
\$HELP COMMAND FULL  
HELP file for system command "FULL"

The "FULL" command restores normal keyboard processing.

FULL cancels the effect of commands "flip" and "fold".  
Upper and lower case keystrokes will be as typed.  
(see also: "flip" and "fold")

Syntax: "FULL"

Minimum size: "F"

### BugNote

Abstract Systems BugNote 20 May 10, 1983

On Exec/95, Gfid has a bug in the default path. The default path can be defined as # (itself) by the Exec command # without reporting an error. When the default path is set to #, Gfid goes into an infinite loop and locks up the system with CTRL-Y disabled. If your system is ENABLED, CTRL-Z, SPJ0403, G will get back to Exec. Otherwise, the only recovery is pushing the LOAD button. I discovered this when I set # to <#<sub by accident. My error was in a command file with a case typo (# vice 3). This bug is corrected in Exec/{A;S}.

### Junk Mail

The Direct Mail Marketing Association (DMMA) maintains a Mail Preference List. That list includes people who do not want to get promotional mailings as well as people who do. Members of the association periodically bounce their mailing lists against the DMMA master list and add or remove names according to customer preference. You can write to the Mail Preference Service, Direct Mail Marketing, Association, 6 East 43rd Street, New York, NY 10017 and ask that you be dropped from

(or added to) direct mail lists. It won't stop all junk mail, but it can stop some.

the questions asked in this issue? Send your answers and requests in. I'd like a little more participation, please.

**Bit Bucket**

Chomping at the bit? One guess where that one comes from. A horse which is chomping at the bit is trying to get the bit in its teeth. It hasn't yet, so is still in control, but is in danger of getting the bit in its teeth and becoming "out of control". One has to keep a tight rein on a horse which is "chomping at the bit". (See Bit Bucket, PL 90/3.) Someone is "chomping at the bit" when he or she is anxious to charge off in some direction, but still under control.

According to Charles Steinhauser, "Adventure" begins when something goes wrong.

**Questions**

What questions would you like answered? Do you have answers? Write and tell us about things of interest to you. How do you use your Poly? Can you find and answer

**In This Issue**

Editorial . . . . .	1
Announcements . . . . .	1
Letters . . . . .	1
Some Observations . . . . .	3
Converting Functions. . . . .	4
HyperText . . . . .	5
Reviews - SuperMicro, TECH Help!. . . . .	6
Advertisements . . . . .	6
Survey Form . . . . .	7
HELP -- fold, flip, FULL. . . . .	9
BugNote 20 - Default Path "#". . . . .	9
Junk Mail - DMMA. . . . .	9
The Bit Bucket . . . . .	10

**Coming Soon**

Poly Meta, More: BASIC for Beginners, PC stuff, System Programmers Notes, Help, BugNotes, Public Domain Software, etc.

**PolyLetter**  
 191 White Oaks Road  
 Williamstown, MA 01267  
 (413) 458-3597

Address Correction Requested



**FIRST CLASS MAIL**

Ralph E. Kenyon, Jr.                      EXP:99#9  
 Abstract Systems, etc.                      184  
 191 White Oaks Road  
 Williamstown, MA                      01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.

PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber ads are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Back volumes of PolyLetter (1980 thru 1989) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (\$3.50, \$4.00, \$5.00).

# PolyLetter



PolyLetter 91/2

Page 1

MAR/APR 1991

## *Editorial*

Politics, n. A strife of interests masquerading as a contest of principles. The conduct of public affairs for private advantage. Ambrose Bierce, "The Devil's Dictionary".

Anyone written to congress lately? The president? From time to time I write on matters that I take a fancy to. Why just the other day I wrote concerning the national (non-)energy strategy. As it happens, my views coincide with the views of my senators on the matter, so they each wrote back long and agreeable letters echoing what I said we needed to do. One of George's flappers, on the other hand, sent me an equally long letter explaining that the things I was complaining about are what I want to do.

Want a good laugh? Write a short specific suggestion on a topic of current interest. Here's one now. Did you know that peanut farmers must have a federal license? And that there are a fixed quota of licenses? The only way someone can become a peanut farmer now-a-days is to inherit or buy one of the existing federal licenses. (I kid you not, I got this straight from ABC News.) Not only that, but it is illegal to import peanuts! We lost nearly half the last crop of peanuts and the price of peanut butter is going through the roof because of this half a century old law. Some schools are dropping peanut butter sandwiches from the menu due to the price increase. Imagine, kids without peanut butter sandwiches! Anyway, here's something good for a laugh. George Bush is pushing for free trade and deregulation as much as possible. Why don't you write and ask him why he doesn't do away with the communist system which governs peanut farming in the U.S.?

Speaking of peanuts, back in 1979 I heard that the Skippy Peanut Butter company

used a Poly-88 to sample the product line for quality control.

In all seriousness, it's good to write and keep your senators and representative informed of your view. To quote one well known senator, "... and that's my view. And if enough of you disagree -- well, I'll change it."

## *Announcements*

American Chemical Diversified, Inc., announced a new computer maintenance kit. The kit contains six different products for computer maintenance. What's different about this kit is that three items have been replaced with ozone layer friendly propellants. The kit does still include two items which contains CFC's (Freon) which is damaging to the ozone layer; however, those items do not yet have replacements available. According to George Brown, the president, these other items are scheduled for replacement as soon as ozone safe replacements become available. George reported that Dupont is planning to provide the replacements by mid 1992 and is building a new plant for just that purpose. Products which can be selected for the kit include a CRT cleaner, hood cleaner, a duster, T.F. solvent (contains freon), Component cooler (contains freon), flux remover, anti-static spray, and lint free wipes. The Kit is \$59.95. Contact George Brown at 202r Parks Avenue, Pelham, NY 10803, 1-800-782-2804.

## *Letters*

Dear Ralph,

April 11, 1991

I found some humor in your last issue. You mention that PM runs too slow for your liking, and Tom Bucy wrote that PM speeds up his operations by over 700 percent! Just shows how much PCs have improved over the past decade.

I was also amused by Jim Salinger's letter. He wasn't interested in my suggestion to move his Poly programs to a PC, and some of his reasons were "cannot afford downtime" and "do not want to depend on outside service people to fix a PC." Those reasons are exactly why I would recommend moving to a PC. So long as Jim has enough Poly parts to keep his systems running, fine. But one day that will no longer be true; he may wish he had moved to a PC before his last CPU (or video, or disk controller...) dies. When that day comes, there won't be a lot of service people lined up for the chance to fix his Poly.

Also in regard to Jim: I admitted to him that I was unable to fix his hard disk problem, and I congratulate you and him for solving it. That was quite a ride!

You might be interested in a small correction to last issue's Bit Bucket. "Champing at the bit," not "chomping," is the correct phrase. Part of a horse's bridle is the bit in his mouth, and he is "champing" when he bites on it, perhaps impatiently. You might say Poly owners were champing at the [binary] bit in the 1970's, when they acquired some of the first bit-slinging machines that fit on a desktop! Best regards, -- Bob Bybee, Stone Mountain, GA.

[If I had 25 and 33 Mhz 3 and 486 machines also, I'd be happier with the speed of everything! -- Not just PM. Why, WordPerfect might even keep up with the Poly! -- Ed.]

Ralph,

April 30, 1991

I am slowly weaning away from the Poly to a 386 DOS machine and believe there may be other readers who may benefit from the dual resources of Poly/PC and "bridges" like PM to get the most out of both. I will send an article. -- Earl Gilbreath, Savannah, Georgia.

### **Back Issues**

How about a PolyLetter back issue clearance sale? PolyLetter has conducted an inventory of back issues and finds that there are plenty of some issues. Here's your chance to get some back issues of PolyLetter at a really reduced rate. The sale price is \$1 each (shipping included). The following issues are included in this

sale.

1980 - 1, 2, & 5  
 1981 - 4, 5, & 6  
 1982 - 4, 5, & 6  
 1983 - 2, 4, 5, & 6  
 1984 - 2, 3, & 5  
 1985 - 2, 3, 4, 5, & 6  
 1986 - 1

1980/1 topics: Protect Part of Your Screen, Machine Language Programming, For Programmers Only, Medical Programs Anyone?, DSDD Mass Storage - 88-MS, BASIC DEF FN Eliminate Cursor, User Input Program Review - Form.OV, & The Ultimate Program.

1980/2 topics: Review of DisAsmb.BS, Screen Layout form, Program Top of Memory Is, Magnetic Disk Holder, HELP boot, & DOM-APR-80 listing.

1980/5 topics: CP/M is Now Available, The C. A. Thompson Method, EVAL explained, Printer POKEs, HEX to decimal conversion, Using Exec from BASIC, How BASIC Stores Variables, Operate Your Poly Remotely, TI 820 Printer Interface, Hash Coding, SORT TIP, & ASCII CHART.

1981/4 topics: CP/M from Poly, Super Pilot, Hard disk - 88/HD, More on cleaning MS drives, Spooler works with Exec/93, WordMaster II hint, BASIC function MOD, Super Zip keyboard speedup, Programs FLIES.BS, INPUT.BS, COUNT.GO, & READ.GO descriptions, DOM-JUL-81 discription, Virtual Universe Operating System, Program PERMUTATIONS.BS, MS Error messages, Dealer Written Software, & Escaping the Front Panel.

1981/5 topics: CP/M Corner, Poly-88 Users Group, Digital Research Information, Edit 4.1.0, MACRO-85, IMSAII, ALTAIR, S-100 boards in the Poly, Dictionary for the Poly, How Computers can make you rich, Stopping Scrolling displays, Letter Formatting in BASIC, & On the FORTH day of ....

1981/6 topics: Bug in PACK - Exec/90, More on Freezing the Screen, INIT system disk, IMAG - remove system disk, Programs CLEARNEW.GO, SETNEW.GO, BACKUP.GO, Sex-Appeal.BS, MOON-LANDER.BS, ARTIL.BS, BACKGAMMON.BS, SLOT.BS, SCOPY.GO, COMP-DISK.GO descriptions., Helpful Hints in Layman's Language, DOM-NOV-81 description, Using Super Zap, REENTRY made

easy, & How to erase yourself.

1982/4 topics: Front Panel Lockout, Blinking Load Light, CP/M Corner, Helpful Hints in LayMan's Language, Microline 80 Printer, Smith Corona TP-1 Printer, DOM-JUL-82 listing, Programs DX.GO, BOWLING.BS, MASTERMIND.BS, CHANGE.GO, TEXT-TRAN.BS, FPL.GO, FIND.GO, & SDIR.GO descriptions, Squeezing Space in BASIC, MS Unit Types, & Using FORMAT from Exec.

1982/5 topics: CP/M Corner, Poly Users on the Source, More on Using Format from Exec, Is Your Poly Covered?, Diablo Habits, Name That Disk, The "System" Disk, & Poly -- Four Years Later.

1982/6 topics: The Stack, Cache.ZO Explained, Adding Mass Storage to Polys, How about "REAL" CP/M, Chess.GO Fix, CP/M Program Exchanges, Portability, & Macros..

1983/2 topics: CP/M Corner, Why an HD/18, DOM-MAR-83 listing, Bug in BASIC ASIN, Bug in ON ERROR, Programs DATA-ENTRY.BS, BASES.GO, SNIFFALL.GO, and SCAN.GO descriptions, & SA-400 Mini-Disk Drive Part 2.

1983/4 topics, CP/M Corner, DOM-JUL-83 listing, Programs DIS80.GO, ERROR.GO, & MKDIR.GO descriptions, Perfect Calc, Upgrading an 8810 to Two Drives, How FORMAT.GO Works, Helpful Hints in Layman's Language, Exec/96, BASIC C04, Using Two Printers from BASIC, {lpi n}, {cpi n}, & {tabs n}.

1983/5 topics: Using FTP as a terminal, Report on the HD/18, Load Addresses and Start Addresses, CP/M Corner, Program PCOPY.GO description, & Modems and Communications.

1983/6 topics: Syquest Bugs Fixed, New ROM for 88-MS, More on Macros, Modems and Communications Part 2, & Disk Failure Modes.

1984/2 topics: Pascal Pastures, Table-Driven Programs, What Do You Say to a Dead Poly?, Chip Seating, Cleaning Edge Connector "Fingers", A Two-Drive 8810: Mechanical, & No 8 bits from the Serial Port.

1984/3 topics: ESC-Control-K, What Do You Say to a Dead Poly?, A Two Drive-8810 Modifying the ROMS, Diagnostics, And the

Bit Goes On (8 bit serial), & What's in a ROM.

1984/5 topics: How it Works - The Video Terminal, 96 tpi PROMS, Software Boots, & DOM-SEP-84 description.

1985/2 topics: Disk Controllers -- How it Works, Review of Anchor Signalman Modem, & Big Book of Photo Copier Humor Rev.

1985/3 topics: Disk Controllers - How it Works, Adding a Solid State Disk to Poly, Modernizing your Poly, Take Heart Dear PolyOwner, & BASIC Review.

1985/4 topics, Modernizing your Poly, For Adventure Fans, Adding a Solid State Disk to Poly, Port "0" Serial Card Upgrade, How to use Mailist, & HELP IMAGE.

1985/5 topics: How to use Mailist, Adventure Patches, How Programs Run on the Poly, etc, Adding a Solid State Disk to Poly, & More About Adventure.

1985/6 topics: Helpful Hints in Layman's Language, How to use Mailist -- Preface, Macros, Using Mailist - Part 4, Put ESC into a Text File, Program COPYLINE.BS, How To Use Setup, PolyLetter Subscriber Address List, Helpful Hints In Laymen's Language, BASIC User Defined Functions: To Strip Blanks, To Skip Lines, & To Draw a Border, & Display Time as Hours and Minutes.

Circle desired issues.

80/1,	80/2,	80/5,	81/4,	81/5
81/6,	82/4,	82/5,	82/6,	83/2
83/4,	83/5,	83/6,	84/2,	84/3
84/5,	85/2,	85/3,	85/4,	85/5
85/6,	86/1			

A very few numbers of copies of some of the other issues are also available -- inquire.

### Obituary

Constantin Pavloff, formerly of Richland Washington, is now computing on that big Poly in the sky. Never again will he have to worry about hardware failure or other downtime. "Conny" was a long standing Poly user who programmed in BASIC; he used his Poly for stress analysis, statistics, inventory control, and, to relax, for games. A moment of silence, please, for someone who never gave up on the poly -- to the very end. We'll miss his contributions to PolyLetter.

**Drives Halt!**

Charles Steinhauser posed a programming problem for PolyLetter. He complained that when a disk is removed from the drive while the red light is on the drive continues to spin forever. He wanted a way to stop the drives faster. PC drives don't care.

Although the manual says not to remove the disk when the red light is on, it is safe to remove the disk once the dollar sign '\$' prompt is on the screen. In fact, some Poly programs say to swap disks before the light goes out. MIRROR in one I can think of right off the top of my head.

How come the drive keeps spinning, you ask? The answer has to do with how the ROMS read and write to disk. The short answer is that the ROMS tell how long to keep the drive running by counting the number of holes sensed. You take the disk out and the disk controller no longer senses holes.

So how come Poly designed the Poly's to wait so long before shutting the drive motors off? Well, the old SA-400 drives took two seconds to come up to speed from a stop. To save wear and tear on the heads, Poly allow 5 seconds before shutting down the motors. When imaging a disk, or when doing lots of disk work, the next disk input or output (Dio) request would often come within 5 seconds. If the motors were already running the wait would much shorter (1/3 of a second) and the drives would be saved a stop and a start.

Ok, aside from rewriting the PROMS, how do we get past the time limit and shut the drives down faster? Well, if the PROMS are earlier than version 81, there's no getting around it without updating or rewriting the proms. But if the PROMS are version 81 or any of the (A1S) PROMS (ASROM), getting around the time limit is possible with a small TSR program which hooks into the Dio vector. Poly's version 81 PROMS, as well as all ASROM versions vector calls to Dio through a RAM area known as worm hole 9 (WH9).

The code at Dio reads:

```
Dio  JMP WH9
```

The code at WH9 reads:

```
WH9  JMP D10
```

Since WH9 is in RAM, this code must be installed at boot-up time by the ROMS. D10 is the internal address of the Dio code and varies with the different flavors of the version 81 ROMS and with the various ASROM's.

It is this feature of the version 81 PROMS and ASROMS that allows adding hard disks to the Poly. The device driver code is added as a TSR program and is hooked in by stealing the address in WH9. The new driver code sends ordinary Dio calls back to D10, but takes care of hard disk i/o itself.

Neat huh? Just by vectoring Dio through a RAM area, all kinds of extra devices could be hooked in. Well, we can take advantage of that to fix Charles's problem. In stead of making him wait for the drives to shut down, we can give him "instant gratification".

Let's see, we want Dio to work as before, but we want the drives to shut down faster. I thought of two ways to do this. The first way was to make the call to Dio as usual and then just stop the drives. There is a routine documented in the System Programmers' Guide called 'Dhalt' which stops the single density drives. No waiting at all -- stop right now!

Our new TSR program must be installed in WH9, but it must do Dio as before, however after Dio is done, it must halt the drives. WH9 now becomes a jump to our new routine.

```
WH9  JMP STOPEM
```

Our new routine must do Dio as before, but it can't use Dio itself, or an infinite loop will result. STOPEM must call the internal address of Dio which is D10. Then it must call Dhalt to stop the drives.

```
STOPEM CALL D10
      CALL Dhalt
      RET
```

This is fine if Dio is successful, but not if Dio returns an Error. Dhalt must not be allowed to destroy the error/success flag or the error code. We can accomplish that by stashing the registers while Dhalt is called:

```

STOPEM CALL DIO
        PUSH PSW
        PUSH B
        PUSH D
        PUSH H
        CALL Dhalt
        POP H
        POP D
        POP B
        POP PSW
        RET

```

A call to Dio would go first to WH9, then to STOPEM, then to DIO, where the usual Dio code is, then the drives would be stopped by Dhalt, and finally return to the caller. We users would see the drives stop immediately when the '\$' prompt is back.

All this is fine, but we must have some way to get this program installed. We must find out what DIO in order to use it. We must also find out where STOPEM is going to be when it is loaded up under MEMTOP. This calls for some fancy installation code.

First, how do we find out what DIO is? Well, before our program is installed, the code at WH9 has it.

```
WH9     JMP DIO
```

Since a JMP code takes up 1 byte and the jump to address takes up two bytes, we want the two bytes starting at WH9+1. We can get them with the LHLD opcode.

```
LHLD WH9+1
```

This will get the two bytes that make up DIO from WH9+1 and WH9+2 and put them in the 8080 register HL. Notice that our routine starts with:

```
STOPEM CALL DIO
```

although we don't know what DIO is. Since a CALL instruction takes up 1 byte, the location where DIO goes is STOPEM+1 and STOPEM + 2. The SHLD instruction will do the job nicely. Our installation code becomes:

```
LHLD WH9+1
SHLD STOPEM+1
```

Now that we have DIO covered, we still have to tell WH9 where STOPEM is. We can do that by putting the address of STOPEM in the same place where we got DIO.

```
LXI H,STOPEM      ;Get our address in HL
SHLD WH9+1        ;Put it in WH9's jump address.
```

All this would be fine, except we want this program to live up under MEMTOP, not where other user programs will run. We won't need to put the installation code up there, but we will need to put the basic routine at STOPEM up under memtop. It would be easy if MEMTOP were the same for everyone, but it's not. MEMTOP will be different depending upon how much memory and how many TSR programs one has. We can find out what MEMTOP is easily enough. MEMTOP is the name of the system variable which store the last usable byte of RAM. We will need to know how much room our routine needs and change MEMTOP to protect the new routine. It wouldn't do to wipe out the code where Dio is going -- anything could happen.

LHLD MEMTOP would get the address of the last usable byte. This is where we need to put the last byte of our new routine. The entire routine only takes 15 bytes. Let's see what we would need to copy the routine up into the proper place.

```

STOPEM CALL DIO      ;3 bytes
        PUSH PSW     ;1
        PUSH B       ;1
        PUSH D       ;1
        PUSH H       ;1
        CALL Dhalt   ;3
        POP H        ;1
        POP D        ;1
        POP B        ;1
        POP PSW     ;1
TAIL   RET           ;1
                                ;15 total

```

We'll write a loop to copy 15 bytes up under memtop. Since we get MEMTOP into HL with "LHLD MEMTOP" we would already have the destination in HL. We can get the source into DE with "LXI D,TAIL". This would leave register BC free for counting. Here's code to copy these 15 bytes up under MEMTOP

```

LHLD MEMTOP          ;Get destination
LXI D,TAIL           ;Get source
MVI B,15             ;Get count
LOOP  LDAX D          ;Get a byte from the source
      MOV M,A         ;Put a byte in the destination
      DCX H           ;Move back to the next destination
      DCX D           ;Move back to the next source
      DCR B          ;Count down
      JNZ LOOP        ;Loop for more if not done.

```

When this code has completed executing HL points at the next lower byte. (The last byte was moved into place and then HL was decremented.) This will be the new last available byte. So, we can restore MEMTOP just by stuffing this new value there.

```
SHLD MEMTOP ;Mark our code protected.
```

Notice that the next byte past MEMTOP is the first byte of our new STOPEM routine. By incrementing HL we get the new value that goes into WH9 to point it at the new location of STOPEM.

```
INX H
SHLD WH9+1
```

This presumes we have first gotten the contents of WH9 for STOPEM's CALL DIO.

Let's put this all together

```
REFS SYSTEM ;Open the system label file.
REF MEMTOP ;Storage for top of RAM
REF WH9 ;RAM vector for Dio
REF USER ;Start of USER memory
REF Msg ;Console message routine.

DIO EQU 0 ;This is unknown to us, but
;the label must be defined.

ORG USER ;This is where we live
IDNT $,$ ;Load and start addresses same

Install LHL D,WH9+1 ;Get the actual value of DIO
SHLD STOPEM+1 ;Put it in its place.
LHL MEMTOP ;Get destination
LXI D,TAIL ;Get source
MVI C,15 ;Get count
LOOP LDAX D ;Get a byte from the source.
MOV M,A ;Put a byte in the destination
DCX H ;Move back to the next destination byte
DCX D ;Move back to the next source byte
DCR C ;Count down
JNZ LOOP ;Loop for more if not done.
SHLD MEMTOP ;Mark our code protected.
INX H ;Move back up to our loaded routine
SHLD WH9+1 ;Install it in WH9
LXI H,Hello ;Tell em were in
JMP Msg ;Let Msg return for us
```

```
Hello DB 'Dhalt installed in Dio.',0DH,0
```

;Our new routine to be copied up under MEMTOP

```
STOPEM CALL DIO ;3 bytes
PUSH PSW ;1
PUSH B ;1
PUSH D ;1
```

```
PUSH H ;1
CALL Dhalt ;3
POP H ;1
POP D ;1
POP B ;1
POP PSW ;1
TAIL RET ;1
;15 total
END
```

Instead of laboriously hand counting the code, we could have devised a way to let the assembler do it for us. The number of bytes, symbolically, is TAIL-STOPEM+1. The above loop copy code works for less than 256 bytes, but won't work when the code to be installed is longer than 255 bytes. The following replacement code will do the job. Replace MVI C,15 with:

```
LXI B,TAIL-STOPEM+1 ;Get count
```

Replace DCR C with:

```
DCX B ;Count down
MOV A,C ;Get our count
ORA B ;Both bytes.
```

Now, when this program is executed, the Dio code will work as before, except that the drives will be stopped immediately afterwards.

But, what happens when you want to image from one drive to another? The drives stop and start between each copy operation. What a bore. Click, click, click... -- tisk, tisk, tisk. Let's take a look at the single density ROM code a little closer and see if we can't get some compromise.

How do the ROMS know how long the drives have been spinning after a normal Dio call? They count the holes passed. In fact, they count down from 250 and, when 0 is reached, call Dhalt. Well, that count must be stored somewhere. Perhaps we can "lie" to the PROMS by telling them that the count is almost up. The count is stored in a location called MTO. The value of the location is 2DA2 hex. This is one variable not included in SYSTEM.SY (although I have put it in mine). We can tell our program where MTO is and, after we call DIO, we can set MTO to something quite small, say 20. 20 is a good number because it is 2 revolutions of the disk. This is only 1/3 of a second, but it gives the drives time to make a worst case cycle during sector by sector drive to drive copy operations.



After waiting 5 seconds, we'd hardly notice 1/3 of a second. To tell the program where MTO is we would need an EQUate.

```
MTO EQU 2DA2H
```

Since we are going to do something simple, like just setting MTO to 20, we won't be calling Dhalt and therefore won't be changing the registers set by Dio. Stopem just has to call DIO and set MTO to 20

```
Stopem CALL DIO      ;3 bytes
        MVI A,20     ;2 bytes
        STA MTO      ;3 bytes
TAIL   RET           ;1
                ;9 total
```

Since the System Programmer's Guide states that A is junk after a call to Dio we needn't bother about A being set to 20 in our new routine. But, if you are a pureist and want the values unchanged we can add a PUSH PSW and POP PSW.

```
Stopem CALL DIO      ;3 bytes
        PUSH PSW     ;1 BYTE -- SAVE IT
        MVI A,20     ;2 bytes
        STA MTO      ;3 bytes
        POP PSW      ;1 byte -- restore it
TAIL   RET           ;1
                ;11 total
```

Let's put this new version all together:

```
REFS SYSTEM ;Open the system label file.
REF MEMTOP  ;Storage for top of RAM
REF WH9     ;RAM vector for Dio
REF USER    ;Start of USER memory
REF Msg     ;Console message routine.

MTO EQU 2DA2H
DIO EQU 0 ;This is unknown to us but the label
        ;must be defined.
ORG USER ;This is where we live
IDNT 9,9 ;Load and start addresses same

Install LHL D WH9+1 ;Get the actual value of DIO
        SHLD STOPEM+1 ;Put it in its place.
        LHL MEMTOP ;Get destination
        LXI D,TAIL ;Get source
        MVI C,TAIL+1-Stopem ;Get count
LOOP   LDAX D ;Get a byte from the source.
        MOV M,A ;Put a byte in the destination
        DCX H ;Move back to the next destination byte
        DCX D ;Move back to the next source byte
        DCR C ;Count down
        JNZ LOOP ;Loop for more if not done.
        SHLD MEMTOP ;Mark our code protected.
        INX H ;Move back up to our loaded routine
```

```
SHLD WH9+1 ;Install it in WH9
LXI H,Hello ;Tell em were in
JMP Msg ;Let Msg return for us
```

```
Hello DB 'Dhalt installed in Dio.',0DH,0
```

;Our new routine to be copied up under MEMTOP

```
Stopem CALL DIO ;3 bytes
        MVI A,20 ;2 bytes
        STA MTO ;3 bytes
TAIL   RET ;1
                ;9 total
```

END

I'll bet you thought this was the end...

One problem with the above program is that it is permanently installed. It does not conform to my TSR memory management policy I discussed in an earlier issue. For this program to be expanded to meet those criteria, it must keep track of whether it is installed or not, and must have some disconnect code. To keep track of whether or not it is installed it need a status byte. This status byte must be checked to see if the program is connected or not and then must disconnect or connect the routine. A nice program would tell the user what the status is as well. This also means that the connect and disconnect code must be up under memtop.

To see if the program is connected, let us use the status byte Zero if not connected and non-zero if connected. The first code must check this byte and then do the proper routine. If we knew what DIO was and where the program was to live, we could use the following code:

```
Begin LXI H,IDmsg ;Our identifier.
        CALL Msg ;Say who we are...
        LXI H,Status ;Point at status byte
        MOV A,M ;Get it
        ORA A ;Check it
        JZ Connect ;Zero, so we gotta connect ourselves.
DisCon XRA A ;Clear our byte.
        MOV M,A ;Set our status to disconnected.
        LXI H,DIO ;Get the old Dio routine
        SHLD WH9+1 ;Put it back where it was.
                ;(cutting us out of the picture).
        LXI H,End ;Get the old Top of RAM
        SHLD MEMTOP ;Restore it
        LXI H,dis ;Say disconnected
        JMP Msg ;Tell 'em and let Msg return for us.

Connect CMA ;Change 00 to FF
        MOV M,A ;Set our status to connected
        LXI H,Begin-1 ;Protect down to here
```

```

SHLD MEMTOP ;Gotta be safe
LXI H,Stopem ;Our new routine
SHLD WH9+1 ;Connect ourselves
LXI H,con ;Say connected
JMP Msg ;Tell 'em and let Msg return for us.

```

```

IDmsg DB '(A:!) Dhalt 1.0 (04/27/91) ',0
dis DB 'dis'
con DB 'connected to Dio.',0DH,0

```

```

Stopem CALL DIO
MVI A,20
STA MTO
RET

```

```

Status DB 0 ;0 for not connected.

```

```

END

```

It turns out that the Poly Assembler knows about relocatable code. We could use most of the above code because Asmb will generate a relocation word map for installing the program. Abstract System's LoadRel.GO program will take a relocatable program and move it up under MEMTOP, fixing up the changed locations.

We would only need to tell Asmb that the above module was to be relocatable by using the RELOC popcode. But, we have a problem in that we don't know what DIO is. We can fix up the connect routine by changing the LXI H,DIO to LHL D WH9+1. But this must happen before Stopem is put into WH9+1.

```

LXI H,Begin-1 ;Protect down to here
SHLD MEMTOP ;Gotta be safe
LHL WH9+1
SHLD Stopem+1 ;
LXI H,Stopem ;Our new routine
SHLD WH9+1 ;Connect ourselves

```

Also, the disconnect routine must be changed. But this time we know where we can get DIO. We can get it just where we put it -- out of Stopem+1. The corrected disconnect code becomes:

```

LHL Stopem+1 ;Get the old Dio routine
SHLD WH9+1 ;Put it back where it was.

```

Being a memory miser, I found an even more efficient way to install and de-install this routine. I noticed that I was changing both the contents of WH9+1 and Stopem+1 and says to myself, why not just swap these and let the status reflect which one is in where? For this to work, I had to start out with Stopem+1 containing

Stopem itself. When I swap these two location contents I needed a temporary storage location, so used the stack. Here's the code and what the locations contain during both connecting and disconnecting. The same routine does both connecting and disconnecting just by putting Stopem into Stopem+1 to start out with.

```

; Code Connecting Disconnecting
Swap LHL WH9+1 ;DIO Stopem
PUSH H
LHL Stopem+1 ;Stopem DIO
SHLD WH9+1 ;Stopem DIO
POP H
SHLD Stopem+1 ;DIO Stopem

```

Here's what my final program ended up looking like.

```

;*****
;# MACRO ;
id DB '(A:!) Dhalt 1.0 (04/27/91) ',0 ;
#L ENDM ;
;# ;
;*****

```

```

REFS SYSTEM
REF MEMTOP
REF Dio
REF MTO ;MTO EQU 2DA2H
REF WH9
REF Msg

```

```

RELOC

```

```

Start LXI H,IDmsg
CALL Msg ;Sign on
LXI H,Status
MOV A,M ;Get it
ORA A ;Check it
CMA ;Reverse it
MOV M,A ;Stow it
DI ;Don't interrupt this switching
JZ $+9
LXI H,End
JMP $+6
LXI H,Start-1
SHLD MEMTOP ;Protect / unprotect
Swap LHL WH9+1
PUSH H
LHL Stopem+1
SHLD WH9+1
POP H
SHLD Stopem+1
EI
JZ $+9 ;Still got that flag set
LXI H,dis
JMP Msg

```

LXI H,con  
JMP Msg

IDmsg id  
dis DB 'dis'  
con DB 'connected.',ODH,0  
Status DB 0

Stopem CALL Stopem ;All that for just this?  
MVI A,20 ;It sure do cost more to  
STA MTO ;administer it than it  
End RET ;does to just do it!

END

When assembled, this program produces a relocatable .RL file which can then be loaded with LoadRel, or it can be converted into a self-relocating .GO file with MakeRel. It satisfies my memory management protocol in that it can be disconnected by ENABLE, ZAP, START. It can also be reconnected by START (in case you really didn't want to remove it after all). I get double duty from the program identification line by putting it into a MACRO. That way I get to see the program identifier at the top of the assembly language source file, but get it put into the correct place when assembled -- no duplication in the text file.

Well, Charles, I hope it was worth it!

### **8" Hard Disk**

Many people know that Poly sold a 5" 18MB drive designated the HD/18. This unit used the Xebec controller and a SASI (Shugart Associates System Interface) controller.

Not so many people knew about Poly's first hard disk unit. It was a PRIAM 8" 10MB drive and used a PRIAM controller. This monster weighs a ton! As if that's not enough, it requires a special power supply which produces 4 different voltages and which also weighs plenty by itself. The drives were actually manufactured for Priam by Hokuskin in Japan. The one I have doesn't even have Priam's name on it. "Hokuskin" is the only name that is printed on the drive and circuit board. In fact, Priam didn't even put their name on the controller card!

Just in case anyone has one of these units, new replacement drives can still be purchased. A little research has located 5

new ones at Sequel, Inc., 2300 Central Expressway, Santa Clara, CA 95054-4972, Phone: 1-408-987-1000. Ask for Donna in Tech Support. But even they didn't find any documentation on these drives.

### **Advertising**

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Noncommercial ads by subscribers are free.

Spring vacation in the Birkshires for sale: One week time-share vacation townhouse at Oak and Spruce Resort, Lee, Mass. -- part a swappable system -- Sleeps 4, 1-1/2 baths, 19th week (starts on mothers' day). Dues paid to 1994 -- priced for quick sale at \$4000. Call 413-354-7750.

500 (count them) 4116 DRAM (16K to 64K upgrade chip) free for the asking: Limit 64 per customer, you pay shipping. Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA 93016-0385

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all) (or swap for something interesting), Doug Schirripa, 716-724-5023 (days) or 716-657-7437. (evening).

Entire PolyMorphic System User Manual, System 88 User's Manual with Exec/96 addendum, & System 88 Operation Essentials On IBM disk. Al Levy, 516-293-8358

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

**DISKS - MODEMS - PROMS - SOFTWARE - SPELL**

1. MAXELL diskettes: 5-1/4" 10 hard sector -- \$10 per box.
2. Used diskettes: 5-1/4" 10 hard sector -- \$0.50 each.
3. Hayes Micromodem 100 (300 baud S-100 internal modem) \$20.  
(If you don't have a modem this is a cheap way to go.)
4. HayesSys modem software (for the Micromodem 100) \$10.
5. Abstract Systems Exec (Enhancements & bugs corrected) \$30.
6. Abstract Systems Proms (Enhancements & bugs corrected) \$35.
7. PolyGlot Library Volumes: \$6 each; 5 or more - \$5 each.
8. Hayes Smartmodem 1200B (IBM compatible internal) \$40.  
Abstract Systems, etc., 191 White Oaks Road,  
Williamstown, MA 01267, Phone: (413) 458-3597  
(Send \$1.00 for a complete catalog--[free with any order].)  
(Make check or money order payable to Ralph Kenyon.)

**Bit Bucket**

Bybee's addendum to Steinhauser's adventure: "'experience' is what you get when you were trying to get something else".

**Questions**

What questions would you like answered?

**PolyLetter**  
191 White Oaks Road  
Williamstown, MA 01267  
(413) 458-3597

Address Correction Requested



**FIRST CLASS MAIL**

Ralph E. Kenyon, Jr.      EXP:99#9  
Abstract Systems, etc.      184  
191 White Oaks Road  
Williamstown, MA      01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.

PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber ads are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Do you have answers? Write and tell us about things of interest to you. How do you use your Poly? Can you find and answer the questions asked in this issue? Send your answers and requests in. I'd like a little more participation, please.

**In This Issue**

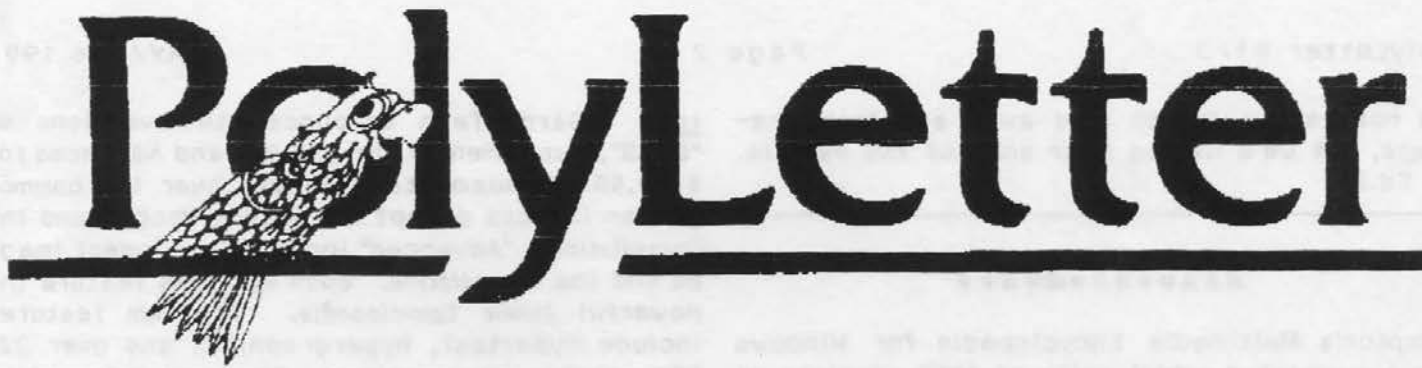
Editorial . . . . .	1
Announcements . . . . .	1
Letters . . . . .	1
Back Issue . . . . .	2
Obituary . . . . .	3
Drives Halt! . . . . .	4
8" Hard Disk . . . . .	9
Advertisements . . . . .	9
The Bit Bucket . . . . .	10

**Coming Soon**

Poly Meta, More: BASIC for Beginners, PC stuff, System Programmers Notes, Help, BugNotes, Public Domain Software, etc.

Back volumes of *PolyLetter* (1980 thru 1989) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (\$3.50, \$4.00, \$5.00).

# PolyLetter



PolyLetter 91/3

Page 1

MAY/JUN 1991

## Editorial

Well, it was probably bound to happen sooner or later. I shudder to admit that I have begun to use the PC more than the Poly. This issue will be the first one formatted entirely on the PC using WordPerfect. I have begun to work seriously on a book-length manuscript in philosophy entitled *Atomism and Infinite Divisibility*. WordPerfect gives me an equation editor and superior graphics capabilities as well as the ability to swap files with my professors. Would you believe not one of them has a Poly!?

The transition to the PC is a painful one. I'm sitting here with two keyboards active at the same time switching back and forth trying to remember which set of commands to use. It's taking its toll. Either I'm getting schizoid or I'm getting bi-lingual. Actually, the problem is a bit more severe than that; I'm using 4 different word processors at the same time, 5 if you count ED-LIN--which nearly nobody does. WordPerfect 5.1 and the WordPerfect Office editor, ED, use almost the same set of commands, but PC Outline and Poly's Edit use very different commands from those. I have found out how to implement several of the Poly editor commands in WordPerfect; I'll tell you more about that later.

My problem is compounded because I had finely tuned format.GO and Edit.GO to take maximum advantage of the flexibility offered by my DataProducts SPG 8050 printer. WordPerfect's printer driver had only been partially implemented and used the basic features in a relatively simple manner. But updating that driver is a story in itself which I'll get to later.

I still use the Poly for much of the writing itself. And there are dozens of programs I run on the Poly for which I have no PC substitute. I haven't yet figured out how to make a telephone call and save the number, date, time, and a comment to a telephone log, or use that log to validate the phone bill when it comes in. Many BASIC programs as well as machine language ones will

need to be adapted to the task. Oh well, no rest for the wicked. As I find out how to do things on the PC I'll keep you informed. (As long as there are still PolyPeople out there who want to know how.) If you find out some trick do write or call us and share it.

## Letters

PolyLetter,

May 11, 1991

I've converted to IBM but still use the Poly occasionally. I need information on changing Poly BASIC programs to DOS BASIC. Thanks, -- Gary Sterling, Hedrick, Iowa.

[Gary, you renewed at just the right time. Your first new *PolyLetter* has a longish article on converting Poly BASIC functions for PC BASICS. -- Ed.]

Dear Ralph,

May 21, 1991

Try to teach an old dog new tricks. That's my answer to Bob's comment on my reluctance to spend the time and money needed to replace Poly.

My age is fast approaching 70, and there is the possibility that my Poly's HD's and spare parts might last longer than I will.

My commodity trading business is on a fast track and the Poly has earned a big bonus in contributing to its success. -- Best wishes, Jim Salinger, Cincinnati, OH.

[We're all closet geriatric cases -- it's just a matter of time before we emerge. Of course, now-a-days, we are all living longer and healthier lives. Oh to live as long by human standards as the Poly has by computer standards! Now there's a lasting thought. (Sorry, I can't resist.) I suspect that there may very well be people interested in how the Poly is used in your commodity trading business. Can you give us an article which describes how the Poly is used?

I'm not asking you to give away any trade secrets, but we'd like to hear some of the details. -- Ed.]

### Announcements

Compton's Multimedia Encyclopedia for Windows for a suggested retail price of \$895 consists of color menus, multiple text and picture windows on the same screen, simultaneously with sound; a "channel-like" user interface (index icons) and advance searching capabilities. Other features are: the entire 26 volume print version on one CD-ROM disk, featuring 9 million words, 32,000 articles, 15,00 pictures and diagrams, 60 minutes of sound, 45 animated sequences and the complete, on-line Merriam-Webster Intermediate Dictionary. The system requires an IBM-AT or 100% compatible; 2MB RAM; Windows 3.0 application; 512K VGA Card or Multisync Monitor; a hard drive; DOS 3.3 or later; and a CD-ROM drive with a controller card. Contact Lynn P. Batts, Britannica Software, Inc., 345 Forth Street, San Francisco, CA 94107, 415/597-5567.

Old Saybrook, CT: GEM Technologies Computer Corporation announced the release of its first product, a PC based Ada compiler which operates on the DOS platform. The product, known as "DeskTop" Ada, began shipping in April. The president of GEMTech, Robert Gruder, stated "The time has come for a company to bring to the public an affordable yet high quality Ada compiler which will permit the end user to perform high quality productions of Ada programming without having to purchase any add-ons."

DeskTop Ada includes a validated Ada compiler, an editor, full mouse support, and a graphics package among its features. The compiler requires 640K of ram and a 5Meg hard drive to operate. DeskTop Ada is priced at \$149.95 and comes with a 30 day satisfaction guaranty and 90 day defection guaranty. [Does that mean it won't leave the country? -- Ed.]

Time Magazine to Publish "Desert Storm" on CD-ROM. New York, NY, -- TIME magazine will pioneer a new form of publishing with "Desert Storm: the First Draft of History", a multimedia magazine on CD-ROM. The disc, designed to be used with a personal computer, was scheduled to be published April 19, 1991. Contact Robert Pondiscio (212) 522-5196 or Linda Rich (818) 955-9999.

Computer program helps Gardeners with pest con-

trol. GardenTech announced two versions of "BUGS", fundamental for \$89.99, and Advanced for \$129.95. Fundamental features over 185 common garden insects except the insect images and the *Consultation*. "Advanced" includes the insect images and the *Consultation*. Both versions feature the powerful *Insect Encyclopedia*. Program features include hypertext, hypergraphics, and over 225 PCX graphic illustrations. For more information Contact Randall Farrar, 1730 Goodman Avenue, Redondo Beach, CA 90278, (213) 372-5810.

OUT OF YOUR MIND... AND INTO THE MARKETPLACE™, 13381 White Sand Drive, Tustin, CA 92680, (714) 544-0247 announces **Automate Your Business Plan**. This package takes the approach developed in the book, *Anatomy of a Business Plan*, and combines it with a text editor and spreadsheet program with predefined planning outlines, spreadsheets, templates, and instructions to provide a working environment for developing personalized business plans. "Completely menu driven". Software: \$79, Software and book \$95.

PKWare™ introduced PKLITE™, an executable File Compressor. [Phil Katz has done it again.] PK-LITE is a compression program that allows IBM-compatible .COM and .EXE files to be significantly reduced in size. The compressed program files can be executed with no "decompression" steps required of the user. Files compressed by PKLITE typically use 40% less disk space. Versions of this software is available in versions for single users or professional distributors. PKWARE also has a Data Compression Library. Contact PKWARE, Inc., 9025 North Deerwood Drive, Brown Deer WI, 53223 Voice: (414) 354-8699, BBS: (414) 354-8670, FAX: (414) 354-8559

### Back in the USSR

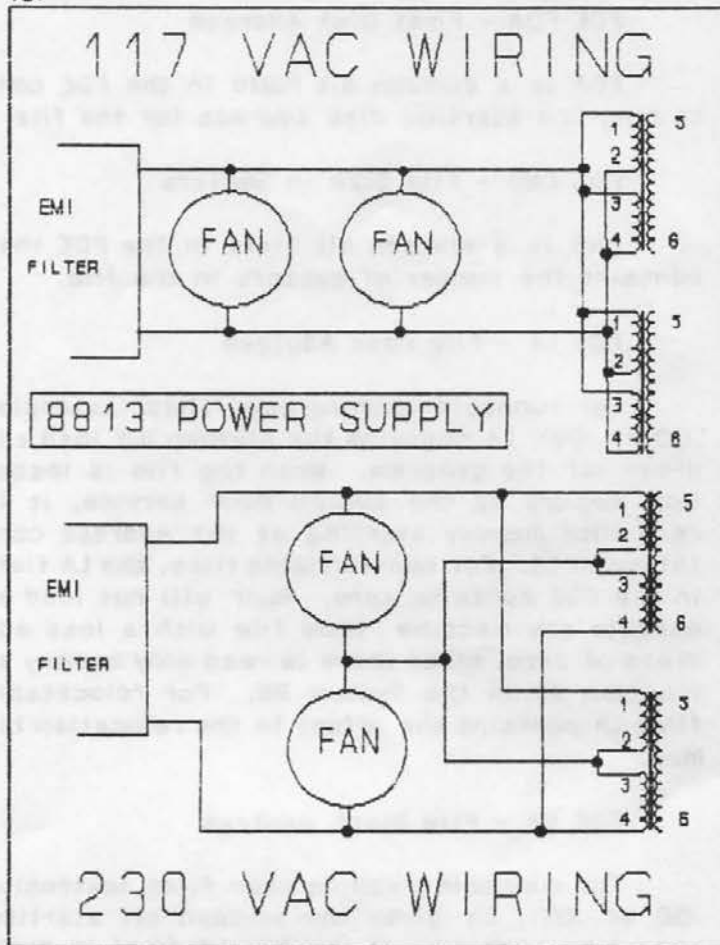
One of our dear beloved Poly users has decide to take a trip to live in the Soviet Union for awhile. She is hoping to participate in the formation of grass-roots democracy movements there. She wanted to take her Poly with her and wanted to know about the power. Most electrical power in eastern Europe is 240 Volts at 50 Hertz. Can the Poly safely eat this stuff?

It turns out that the Poly was designed with just such emergencies in mind. The existing power supply can be rewired for these voltage conditions. The real-time clock will run 16% slower, but this won't hurt anything that depends upon it. The disk drives will take longer to stop. The confidence package will report serial

baud rate errors because it uses the real-time clock interrupts for checking the baud rate transmission.

My friend tells me that the Soviet computer literacy is so low that the mere appearance of the Poly could arouse the KGB. "This has gotta be a super computer! Da!" No one would believe it's a dinosaur.

Well, I looked in my files and found Poly's schematic for the 240 Volt power supply. Here it is:



**New Tricks**

Who says you can't teach the experts (old dogs new tricks)? Al Levy just called me. He was all excited about Poly's BACKUP and REBUILD programs. For years he has been using IMAG and COPY or Scopy to make backups of his not perfectly reliable 8" disks. He just tried Poly's BACKUP and discovered that it breaks large files up into parts on small disks. Naturally, REBUILD puts the file back together again! Talk about excitement! BACKUP copies files with the new bit set to the destination. It skips files with the new bit clear. In this manner it only backs up

files which have been modified or created since the last time it was run. BACKUP automatically clears the new bit when the file has been successfully copied. [For more information on the new bit, see the article on File Directory Entries below.

Al tells me that roughly the same thing can be done on DOS using the XCOPY program. First you must set the Archive bit on the files to be backed up. Then you use "XCOPY source destination /M" which tells XCOPY to copy only those files which have been modified. The /M switch causes the archive bit to be cleared once the file has been copied--just like Poly's BACKUP and the new bit.

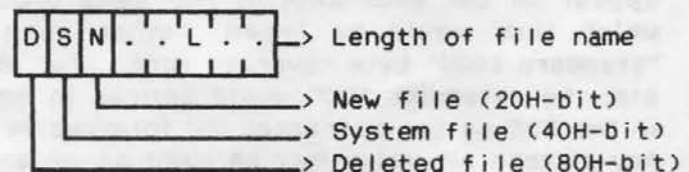
**System Programmers Guide**  
File Directory Entries (FDE's)

The File Directory Entry (FDE) defines a file on the disk. The FDE contains all the information required to locate, access, and delimit the file data on the disk. Since the file name in the FDE is of variable length, the FDE itself is also of variable length. The FDE consists of the following information (in this order within the FDE):

- Flag byte (8 bits)
- File name (variable length)
- File extension (16 bits)
- FDA - Starting disk address (16 bits)
- DNS - File length in sectors (16 bits)
- LA - File load address (16 bits)
- SA - File start address (16 bits)

The FDE format (in a slightly modified form) is used by the system Gfid utility for looking up and entering file names into the directory.

FDE Flag Byte: The first byte of the FDE contains three one bit flags (D, S, and N) and five bits for the file name length:



The 80H bit (D above), if set, indicates that the file has been deleted. If this bit is set in an FDE, that FDE will not be examined in the file look-up procedure and will not be displayed by the system LIST command. FDE's marked deleted are returned to normal status by the UNDELETE

command or the ARISE program. The space taken up by deleted files, both in the directory area and on the disk, is reclaimed by the system PACK command.

The 40H bit (S above), if set, denotes a "System" file. System commands such as DELETE, TYPE, RENAME, and PRINT check this bit. A file marked by the System bit may not be deleted, renamed, edited, or displayed by PRINT or TYPE.

The 20H bit (N above) denotes a "new" file. When a file is created or changed, its corresponding FDE is marked with the "new" bit to make it eligible for saving by the system file maintenance processor, BACKUP, which then clears the new bit. Any combination of the above three bits is allowed.

The last five bits of the flag byte give the length of the file name that follows the flag byte. This restricts the file name to 31 characters or less (a file name must be at least one character long). Note that the file name length DOES NOT include the two character extension.

#### FDE File Name

The file name follows the FDE flag byte and is the only variable length entry in the FDE. The number of bytes used by the file name is contained in the lower five bits of the FDE flag byte. File names usually consist of seven bit ASCII characters, although programs may generate file names consisting of arbitrary eight bit quantities that cannot be entered from the keyboard. When a file name is displayed on the screen, control characters (ASCII 00 to 1FH) appear as Greek characters.

#### FDE Extension

The FDE file extension is a sixteen bit (two character) field that follows the file name. The extension identifies the type of file. The bytes appear in the extension in the same order in which they would be typed, rather than the "standard 8080" byte reversed form. For example, the extension "GO" would appear in memory in the FDE as the character "G" followed by "O". Any sixteen bit value may be used as an extension. A number of extensions are predefined and recognized by the system (as the system expands, this list may also expand):

<u>Extension</u>	<u>Use</u>
DX	Sub-Directory
GO	Runnable machine code file

OV	System overlay
BS	BASIC source program
DT	BASIC data file
TX	Text
SY	Symbol table file
RL	Relocatable machine code file
ED	Editor key definition library
FM	Form File
FV	Form Values File
FW	Form Work File
ZO	On-board code for DSDD Controllers
DD	Device Driver for Hard Disk

#### FDE FDA - First Disk Address

FDA is a sixteen bit field in the FDE containing the starting disk address for the file.

#### FDE DNS - File Size in Sectors

DNS is a sixteen bit field in the FDE that contains the number of sectors in the file.

#### FDE LA - File Load Address

For runnable machine code files (extension .GO or .OV), LA contains the sixteen bit load address for the program. When the file is loaded into memory by the system Runr service, it is read into memory starting at the address contained in LA. For non-runnable files, the LA field in the FDE contains zero. Runr will not load or execute any machine code file with a load address of zero, since there is read only memory at location 00 in the System 88. For relocatable files LA contains the offset to the relocation bit map.

#### FDE SA - File Start Address

For runnable machine code files (extension .GO or .OV), SA gives the sixteen bit starting execution address. If the FDA LA field is zero, indicating a non-runnable file, this field may be used for other purposes.

---

### **Converting BASIC**

One significant difference between Poly BASIC and PC BASIC's is the way they handle data files. The Poly stores strings of text and numbers in ASCII format. An ASCII 13 (carriage return) is used as a file record delimiter. Early Poly BASICS were a little forgiving in this regard. A carriage return in the file could be written over by other data, but BASIC C04 shows no mercy. Once a carriage return is written to



a data file, it's there to stay! No amount of INOUT file processing can remove one.

PC BASICS are somewhat tricky about both commas and quotation marks. PC BASIC's expect sequential file records to be separated by commas and like strings to be in quotes.

For example, if the file contains

```
Williamstown, MA
```

then `INPUT#1,A$,B$` will get **Williamstown** into `A$` and **MA** into `B$`.

But if the file contains

```
"Williamstown, MA"
```

then `INPUT#1,A$` will get **Williamstown, MA** into `A$`.

PC BASIC sequential files report an end of file error when one tries to read past the end of the file; Poly BASIC merely returns a string of length 0 (""). In PC BASIC one must test for the end of file **before** trying to read from the file.

<u>Poly</u>	<u>PC</u>
<code>INPUT:4,A\$</code>	<code>IF EOF(1) THEN 100</code>
<code>IF A\$="" THEN 100</code>	<code>INPUT #1,A\$</code>

Random access files are apparently not so limited. These files work much like Poly fixed length record INOUT files. Actually, the ability to indiscriminately write numbers to Poly files has caused no end of programming problems. If a file was created by writing an integer between 10 and 99 into the file, the data record used 3 characters plus a carriage return. If you tried to write the number 123 to the same position later, the "3" would be lost and only " 12" would actually get into the file. No error would be reported.

We programmers soon learned to figure out what the largest size a number could be, convert it to a right justified string using the function `STR$(N,%#NI)` and to dimension the string to the right size. We would create the data file with enough space for the largest sized numbers and with fixed length records. The best way is to dimension a single input/output string to the size of the record and then to use the function `MID$` to pick out the appropriate portions of the record.

For example, a file with records consisting of a 20 character name, a 4 digit price figure,

and a 7 digit phone number might be dimensioned as follows

```
10 DIM N$(1:20),P$(1:4),T$(1:7),I$(1:31)
20 REM I$=N$+P$+T$
```

The `I$` variable would be used for both INPUT and OUTPUT.

For INPUT we do the following:

```
30 FILE:4,POS,X
40 READ:4,I$

50 N$=MID$(I$,1,20)
60 P$=MID$(I$,21,4)
61 P=VAL(P$)
70 T$=MID$(I$,25,7)
```

Now we can print the name, `N$`; Phone #, `T$`; and amount, `P` whenever we wish.

For OUTPUT we do the following.

first we must make sure the strings are "full up" to their maximum length. This will make sure that when we put them together they will end up in the correct positions in the file.

```
100 N$=N$+" "
110 P$=STR$(P,%#4I)
120 T$=T$+" "
130 I$=N$+P$+T$

140 FILE:4,POS,X
150 WRITE:4,I$
```

GW BASIC does not handle dimensioning of variables in the same way that Poly does, so this scheme won't work exactly the same way. In Poly BASIC we dimension a string variable to the length we want for the data file record length. In GW BASIC the file record length is set by the statement that opens the file.

<u>Poly:</u>
<code>10 DIM N\$(1:20),P\$(1:4),T\$(1:7),A\$(1:31)</code>
<code>20 FILE:4,OPEN,"DATA",INOUT</code>

<u>PC</u>
<code>10 OPEN "R",#1,"DATA",31</code>
<code>20 FIELD #1, 20 AS N\$, 4 AS P\$, 7 AS T\$</code>

The `FIELD` statement does the work of both the Poly DIM statement and the assignment statements.

To input data on the PC one must use the

**GET** statement.

```
GET #1,X
```

X is the record number in the file, the same as the Poly POS number.

```
FILE:4,POS,X \READ:4,A$
```

When the **GET** statement is executed, N\$, P\$, and T\$ are all automatically filled up with the data from the file. There no need for the MID\$ conversion. But this is not always convenient and we could define the FIELD statement similar to the Poly's DIM statement.

```
20 FIELD #1, 31 AS A$
```

Then we would have to do the MID\$ thing, but it would be different from the Poly. This could actually make converting between Poly and PC BASIC programs easier. But we must remember that the MID\$ function is different between both BASIC's. In Poly BASIC it's:

```
MID$(string,starting position,ending position).
```

In PC BASIC it's:

```
MID$(string,starting position,number of characters).
```

where the **number of characters** is optional.

Like Poly's BASIC, if there aren't enough characters the function will return an empty string. But when the number of characters is omitted, then the function will return the rest of the string.

Suppose the Poly MID\$ function is setup as follows

```
MID$(A$,X,Y)
```

The converted function is

```
MID$(A$,X,Y-X+1)
```

There is another important difference between Poly and PC BASICS concerning the MID\$ function. In Poly BASIC, strings are dimensioned to the number of characters. PC BASIC allows the size of a string to be flexible, up to 255 characters. The result is that adding 20 spaces to a string will make it 20 spaces longer! But if the resulting length is greater than 255 an error will result.

Poly discards the extra characters. A little experimenting suggests that the FIELD statement limits how many characters get written out to the data file. If the strings are concatenated by

```
I$=N$+P$+T$
```

the extra spaces will cause P\$ and T\$ to be in the wrong place and maybe cause data to be lost. Suppose N\$="John James Brown", P=2235 and T\$="4588421". When the code

```
100 N$=N$+"
110 P$=STR$(P,%#41)
120 T$=T$+"
130 I$=N$+P$+T$
```

is run on the Poly, I\$ will be:

```
John James Brown      22354588421
```

But when the following code is run under PC BASIC

```
100 N$=N$+"
110 P$=STR$(P)
120 T$=T$+"
130 I$=N$+P$+T$
```

I\$ will be:

```
John James Brown      22354588421
```

Before the PUT statement can write the data to the file, a LSET statement must put the data in the buffer defined by the FIELD statement. This can be accomplished by changing line 130 above to

```
130 LSET I$=N$+P$+T$
```

LSET insures that the data is left justified in I\$ and is in the buffer defined by the FIELD statement. Once the data is in the buffer, it can be written to the file with the PUT statement. In the above example, the PUT statement will cause the following to be written out to the file:

```
"John James Brown      "
```

I\$ will still be:

```
John James Brown      22354588421
```

But only the first 31 characters will be written to disk.

We can correct for this but we must first look at the STR\$ and VAL functions. The VAL functions works similar, but the Poly VAL function reports an input error if the string is not a valid number. The PC BASIC function just returns 0. If you trap the error with an ON ERROR statement, you won't need that statement in the converted program.

The STR\$ function in the PC is more limited than in the Poly. It can't perform the formatting function the Poly STR\$ function does. You'll need to format the number first and then convert it. For example X\$=STR\$(X,%#5F2) does in one step what PC BASIC requires the following to do.

```
10 X$=STR$(INT(100*X+.5)/100)
20 X$=MID$(X$,2)
30 IF LEN(X$)<5 THEN X$=" "+X$:GOTO 30
```

Both STR\$ and VAL functions must be converted with care. Poly's STR\$ function automatically rounds to the nearest whole digit.

Most of Poly's formatted print statements can be converted to PC "PRINT USING" statements.

```
PRINT %5F2,2/3
```

will give " .67", rounding the fraction into the chosen number of decimal points. PC BASIC's PRINT USING statement will also round correctly.

```
PRINT USING "##.##";2/3
```

correctly gives " .67".

But the PRINT USING statement cannot be used for converting numbers to strings in formatted form as Poly's STR\$ function can. The numbers must first be rounded to the correct precision. Here are some rounding functions needed for PC BASIC.

```
0 decimal points  X=INT(X+.5)
1 decimal points  X=INT(X*10+.5)/10
2 decimal points  X=INT(X*100+.5)/100
3 decimal points  X=INT(X*1000+.5)/1000
4 decimal points  X=INT(X*10000+.5)/10000
5 decimal points  X=INT(X*100000+.5)/100000
```

Once the number are rounded, they may be converted with the STR\$ function. However spaces may need to be added or deleted to make the length correct. Poly's STR\$ function without any formatting instructions always adds a blank to the front. The PC STR\$ function only adds a leading blank to positive numbers. Strings con-

verted from negative numbers have a leading minus sign. On the PC,

```
STR$(1) gives " 1"
STR$(-1) gives "-1"
```

Now we can talk about converting MID\$ statements for PC RANDOM files (Poly INOUT files). Input is fairly easy, but output requires more processing. Here's a program which illustrates both input and output to a RANDOM file.

```
10 OPEN "R",#1,"DATA",31
20 FIELD #1, 31 AS A$
REM Input is fairly easy.
25 REM READ FROM THE FILE
30 GET #1,X
50 N$=MID$(I$,1,20)
60 P$=MID$(I$,21,4)
61 P=VAL(P$)
70 T$=MID$(I$,25)
75 REM Do something with the data
76 PRINT N$;
77 INPUT "new value?",W$
78 IF W$<>" " THEN N$=W$
79 PRINT P;
80 INPUT "new value?",W
81 IF W<>0 THEN P=W
82 PRINT T$;
83 INPUT "new value?",W$
84 IF W$<>" " THEN T$=W$
85 Now reformat the data for writing
100 N$=N$+" "
101 N$=MID$(N$,1,20)
110 P$=STR$(INT(P+.5))
111 P$=MID$(P$,2)
112 IF LEN(P$)<4 THEN P$=" "+P$:GOTO 113
120 T$=T$+" "
121 T$=MID$(T$,1,7)
130 I$=N$+P$+T$
140 PUT #1,X
```

Another strange thing about PC BASIC is that you won't get an error when you try to open a non-existent file. BASIC will create a file with no data in it.

PC BASIC offers another way to store numbers than in ASCII format. Numbers can be converted into non-ascii strings with the MKI\$, MKS\$, and MKD\$ functions.

```
MKI$ converts an integer into a 2 byte string.
MKS$ converts a single precision number into a 4 byte string.
MKD$ converts a double precision number into a 8 byte string.
```

Such strings must be converted back before they can be used as numbers. The CVI, CVS, and CVD functions do the job. One advantage of this scheme is that these forms of storage take up less file space than standard ascii numeric values. Here's how the revised program will look.

```

10 OPEN "R",#1,"DATA",31
20 FIELD #1, 31 AS A$
REM Input is fairly easy.
25 REM READ FROM THE FILE
30 GET #1,X
50 N$=MID$(I$,1,20)
60 P=CVS(MID$(I$,21,4)) <---
61 <---
70 T$=MID$(I$,25)
75 REM Do something with the data
76 PRINT N$;
77 INPUT "new value?",W$
78 IF W$<>"" THEN N$=W$
79 PRINT P;
80 INPUT "new value?",W
81 IF W<>0 THEN P=W
82 PRINT T$;
83 INPUT "new value?",W$
84 IF W$<>"" THEN T$=W$
85 Now reformat the data for writing
100 N$=N$+" "
101 N$=MID$(N$,1,20)
110 P$=MK$$(P) <---
111 <---
112 <---
120 T$=T$+" "
121 T$=MID$(T$,1,7)
130 LSET I$=N$+P$+T$
140 PUT #1,X
150 GOTO 30

```

There is another strange thing about opening a non-existent RANDOM file. If you enter a record with data, then the garbage that was on disk or in memory becomes the data for earlier records that had not yet been written.

Well, I started out talking about the differences between the way PC BASICS and Poly BASIC handle files. That has led to talk of string variables, dimensions, and data type conversion. As you can see, there really isn't any nice direct formula for converting data file handling statements from Poly BASIC to PC BASICS. The task requires understanding how the two different BASICs handle data and designing conversions which take advantage of the strengths of each system. I'll get into this more in the future; don't hesitate to call me to research any particular example. And if you find out any special tricks, by all means share them with us.

## Advertising

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Non-commercial ads by subscribers are free.

---

Spring vacation in the Berkshires for sale: One week time-share vacation townhouse at Oak and Spruce Resort, Lee, Mass. Sleeps 4, 1-1/2 baths, 19th week (starts on mothers' day). Dues paid to 1994. - Part of a swappable system. - Priced for quick sale at \$4000. Call 413-354-7750.

---

500 (count them) 4116 DRAM (16K to 64K upgrade chip) free for the asking: Limit 64 per customer, you pay shipping. Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

---

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA 93016-0385.

---

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all or swap for something interesting), Doug Schirripa, 716-624-370 (days) or 716-657-7437 (evening).

---

PolyMorphic System User Manual, System-88 User's Manual with Exec/96 addendum, & System-88 Operation Essentials On IBM disk. Al Levy, 516-293-8358.

---

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

---

## DISKS - MODEMS - PROMS - SOFTWARE - SPELL

1. MAXELL diskettes: 5-1/4" 10 hard sector - \$10 per box.
  2. Used diskettes: 5-1/4" 10 hard sector - \$0.50 each.
  3. Hayes Micromodem 100 (300 baud 5-100 internal modem) - \$20. (If you don't have a modem this is a cheap way to go.)
  4. HayesSys modem software (for the Micromodem 100) - \$10.
  5. Abstract Systems Exec (Enhancements & bugs corrected) - \$30.
  6. Abstract Systems Proms (Enhancements & bugs corrected) - \$35.
  7. PolyGlot Library Volumes: \$6 each; 5 or more - \$5 each.
  8. Hayes Smartmodem 1200B (IBM compatible internal) - \$40. Abstract Systems, etc., 191 White Oaks Road, Williamstown, MA 01267, Phone: (413) 458-3597 (Send \$1.00 for a complete catalog--[free with any order].) (Make check or money order payable to Ralph Kenyon.)
-

**BugNote**

Abstract Systems BugNote 021.0 May 10, 1983

Exec/95 Gfid has a bug in the enter function. Calling Gfid with the Enter function (A=1) returns the incorrect error when the file to be entered already exists. Gfid returns Error 0505: I can't write: the disk is full. The correct one is Error 0600: That file already exists. This bug is subtle, because most programs which use Gfid report the error back to the user when Gfid returns the error. If you have a program which creates output files, and get Error 0505: I can't write: the disk is full, the problem may be the output file already exists.

**Survey-Interview**

A couple of issues ago I sent out survey forms. Many of you have not returned the form so I really haven't enough data to do any kind of reliable statistical analysis. We did get about 10 times the normal response rate compared to other magazines. What I've decided to do is to put the survey questions and answers together and call it a *Pseudo-interview*. Our first such "contrived" interview is with Bob Bybee. Bob was the editor of *PolyLetter* for the three years from PL-81/1 through PL-83/6. (That's 1981 through 1983 for those of you who use the Julian calendar instead of the PL calendar.)

PL: "Bob, tell us about yourself. What do you do?"

BB: "I'm a computer hardware/software designer and technical writer, currently working for Innovative Technology Inc., a manufacturer of voice--messaging systems and other interactive voice systems."

PL: "Well, Bob, "For our readers, would you please describe your Poly System?"

BB: "My Poly systems? I've got one 8810 with 2 SSDD drives and one 8813 with 2 SSDD drives and a 5 meg hard disk."

PL: "What do you use your Poly for mostly?"

BB: "Both my Poly's are currently unused except as necessary to support my existing customers who are mostly PM users."

PL: "What other computers do you have?"

BB: "My other computers? I have an original 4.77

MHZ IBM-PC upgraded with a 20 meg HD, a V20 CPU, one 1.2 meg floppy, and one 360K floppy. I also have a 25 MHz, 80386 PC-clone, with an 80 meg HD, one 1.2 meg floppy, and a 1024x768 VGA display."

PL: "That's an impressive display of hardware. What kinds of Poly articles would you like to see in *PolyLetter*? Do you want to see more or fewer DOS related articles?"

BB: "I can't offer a lot of suggestions for articles on either DOS or Poly subjects. Let's face it, the Poly universe has about sucked itself into the black hole of no escape. I admire your keeping the newsletter going, Ralph. I can't say that I would have pursued it this far if it were me."

PL: "Well, Bob, I must admit, I do it as a labor of love rather than for economic gain. It has been an interesting 5 years. But let's get back to you. What kinds of things would you like to have printed in *PolyLetter*?"

BB: "I suggest that you consider quality over quantity. Perhaps you could cut back on the publication schedule, to 4 times per year, and still put out material that would be of interest to the Poly community. I suggest that approach instead of trying to force out 6 issues per year and padding them with DOS material, or re-running old Poly material. There are too many DOS magazines already, so I would advise not printing too much DOS stuff unless it would be especially useful to Poly people."

PL: "Well, *PolyLetter* could stick to nothing but Poly related material; there's certainly enough new topics that *PolyLetter* could continue for years without ever mentioning DOS. I haven't covered all the languages that run under Exec, and I haven't yet done on article on Poly Meta. There's loads more help files and BugNotes. But, as you say, there are some System-88 users out there who are running their system on PM under DOS. Several readers have asked for DOS related stuff, particularly on converting both BASIC programs and COMMAND files, as well as how to do under DOS what they did under Exec. DOS articles aren't used as filler; they're in demand. There are also readers who would like *PolyLetter* to become 100% DOS oriented. But we can't have that, now, can we? After all, this is *PolyLetter*. And for those who might be concerned, I have no intention of cutting back to a quarterly publication. Right now *PolyLetter* is published shortly after the two month period for the issue."

My first priority is to support reader's requests, and to try to meet the real needs behind conflicting requests. The Poly, with its unique operating system, and its classy walnut case, as well as the high quality of its programming, has established a standard few other computer systems have even approached. Few other systems have commanded the loyalty that Poly has enjoyed. I'd like to apply this standard to pick out DOS or other hardware and software items which approach the Poly standard and feature them in *PolyLetter*. But for those who stick with the Poly, I want to continue to support the viability of the system as well as the hardware. Let's share information on applications as well as on service.

**Help!**

(How does it work?)

HELP BASIC FUNCTION VAL

HELP file for BASIC FUNCTION "VAL"

"VAL (string expression)" returns the numerical value of a numeric string if the string expression begins with a valid digit (0-9). VAL stops when a non-numeric digit is found.

```
VAL("45") = 45,
VAL("3P") = 3,
VAL(" ") = "Input error"
```

To avoid errors, use VAL("0"+A\$), but this will return 0 if A\$ contains a leading blank such as is returned by the STR\$ function.

**In This Issue**

Editorial . . . . .	1
Letters . . . . .	1
Announcements . . . . .	2
Back in the USSR . . . . .	2
New Tricks . . . . .	3
System Programmers Guide	
File Directory Entries (FDE's)	3
Converting BASIC . . . . .	4
Advertisements . . . . .	8
BugNote 21 - Gfid . . . . .	9
Survey-Interview . . . . .	9
HELP BASIC FUNCTION VAL . . . . .	10



**PolyLetter**  
 191 White Oaks Road  
 Williamstown, MA 01267  
 (413) 458-3597

Address Correction Requested

**FIRST CLASS MAIL**




Ralph E. Kenyon, Jr.                      EXP:99#9  
 Abstract Systems, etc.                      184  
 191 White Oaks Road  
 Williamstown, MA                              01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.

PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber adds are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Back volumes of *PolyLetter* (1980 through 1990) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (\$3.50,

# PolyLetter



PolyLetter 91/4

Page 1

JUL/AUG 1991

## Editorial

Well, I see by the clock calendar on the machine that it is time for another issue of *PolyLetter*. It seems like the summer has flown by. Psychological time is often measured by the number of events occurring. We sure have had a lot this last month. How about all the changes in the USSR (if there still is one by the time this gets delivered)? Do you think the Russians are ready for Poly's? Did they read the last issue of *PolyLetter*? No, all this change is not the fault of our Poly user and her Poly going there. She hasn't made it yet. But just imagine how much smoother things might have gone had the Poly been able to help (provided the KGB didn't have them).

We have the second in our series of pseudo-interviews in this issue. The man on the spot is Jim Salinger. Also, Poly feedback has asked for more comparisons between the Poly world and the world of DOS. In this issue is an article comparing some extensive file management strategies between Exec and DOS. Sorry, but the Poly Meta article is pushed back again.

---

## Letters



Dear Ralph, July 15, 1991

Replying to your query, there are no secrets to being a successful commodity trader unless

they would be to spend time every day inputting data that the market generates, calculate market possibilities that might occur based on actual performance for a reasonable time period, and developing the technique for using the output which has been generated from the input data.

This process requires me to "work" from

about 8 to 10:30 A.M., and from 12:30 to 1:00 P.M. every day. At least a dozen Poly programs are involved and automatically run in sequence. Orders are FAXed to a broker before the market opens and further orders might be entered at about 12:45 P.M.

Every commodity has its own distinctive trading pattern which must be learned from experience. Most markets seem to be subject to manipulation by the pros, but learning to live with this certainty is not too difficult. Relying on "expert" advice and poop published daily in the *Wall Street Journal* is a mistake--pay no attention to it because it is usually exaggerated and often dead wrong within a very short time after its release.

Even if I were capable of writing trading philosophy or wisdom, I doubt that it would be understood except by people who might have more experience than I have.

Without describing how I use the info that Poly furnishes, I am attaching one set of work sheets that it produced for me on 13 commodities today. Only portions of this data were used to form the trading strategies.

Sorry, but I really do not have the time or the desire to impart more of or discuss details of my activity. Some people do not have the aptitude to fly airplanes, some cannot handle music; to some computers are totally beyond comprehension. I am unable to assess what a person has "going in" except my wife, who, smart lady that she is, is unteachable in this domain (in my opinion).

You asked for this letter. Publishing it is your decision. I appreciate your dedication to Poly and the service you so faithfully provide for and endangered species. So with my sincere thanks goes my apology for not being more effusive. -- Best wishes, Jim Salinger, Cincinnati, OH.

[Jim, Thanks for taking the time to respond. I can make some educated guesses from what you did and did not say. For our readers, let me add the following. Publishing the three sheets of data Jim sent is beyond the scope of *PolyLetter*, but I can say this about it. Jim has programmed the Poly to organize, process, and present both the raw data he feeds it and some statistical analyses of that data. It is presented in a form that is meaningful to Jim. It picks out relations he believes are significant in trading. Why do I say "he believes", you ask? There are many trading philosophies, and Jim alluded to his when he stated that he relies (exclusively?) on actual performance data and that we should ignore the opinions of the experts.

I, personally, am not qualified to judge any particular market philosophy or strategy. But it seems that using the Poly is excellent under Jim's philosophy. The Poly does not process rumor, opinion, or fears; it processes hard numbers. I note that Jim makes use of such "standard" market information as high, low, close, percent change, and a host of other labels whose names I do not know. I also noticed that Jim calculates moving averages and has the Poly tell him what certain ranges above and below level indicators are. Mind you, this is just computing and plotting relations derived from analysis of the raw data. The real secret comes from the experience of knowing how to use that information, and that is experience Jim has built up over the years. If it's like any other "soft" technical expertise, much of it is experienced as a "feel" and is difficult to communicate. As Jim said: "Every commodity has its own distinctive trading pattern which must be learned from experience."

For example, and I am largely guessing about this, suppose you kept track of the high and lows over the last 10 days of market trading. Each day there would be a certain range to the data. One might compute the average "volatility" over that period, take the current average price and add and subtract the average range change to get up and down level indicators. Only experts would have some feel for what these numbers might mean but they can be used to help decide whether to buy or sell and at what price to do so. The real advantage of the Poly is in analyzing the raw data and presenting the relations for someone with experience like Jim to use. -- Ed.]

Feedback.

July 26, 1991

Keep the focus on computer subjects, specifically on the Poly. Present applications of the system. I enjoy comparisons of BASIC, Exec, hardware, etc., with other systems. Your "Pseudo-interview" is an excellent idea. -- Ken Lowe, West Valley City, Utah.

[Ken, I'd like to do one on you. How about sending your survey form in? -- Ed.]

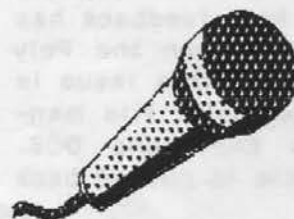
Feedback

August 15, 1991

I use a 386/20Mhz clone daily; I haven't powered up my Poly in 2 1/2 years. I'm not a computer whiz--I'm just "loyal" and believe I can learn something of esoteric and possibly practical value from the communications of this unusual pool of subscribers. -- Robert Stricklin, Warrington, Oregon.

### Survey-Interview

This is the second in a series of *Pseudo-interview's*. Our current "contrived" interview is with James Salinger. Jim has been a frequent contributor to the spirit of sharing that *PolyLetter* likes to encourage.



PL: "Jim, tell us about yourself. What do you do?"

JS: "I'm 69 years old and I jog about 35 miles a week. I'm an ex U.S. Naval aviator and was employed by U. S. Shoe Corporation from 1945 to 1965. Since then I've been a self-employed business consultant and investment advisor."

PL: "I was in the Navy myself, Jim, but in submarines. That's how I first got started with computers and where I was when I got my Poly. Speaking of our favorite computers, Jim, would you describe your Poly Systems for our readers?"

JS: "I have four 8813's with 3 drives. Each one has a Poly hard disk. Two of the hard disks are ST-225'S, one is a Rodime, and the last one is a Nec. For printers I have one Epson 100 and two Brother's -- an HR-35 and an HR-40."

PL: "That's the most powerful array of Poly



equipment I've heard anyone has to date. What do you use all that Poly hardware for mostly?"

JS: "It's used for business planning and in the preparation of investment advice, particularly commodity trading. I designed and wrote the programs myself. I also use it for word processing."

PL: "Your letter about commodity trading will appear in the same issue as this 'interview'. Interested readers should be sure to check out your comments there. That's a story in itself. What other computers do you have?"

JS: "I have a Visual - a cheap PC clone."

PL: "What kinds of Poly articles would you like to see in *PolyLetter*?" What kinds of things would you like to see us print?"

JS: "*PolyLetter* is of general interest. I really can't suggest articles on specific subjects."

PL: "Do you want to see more or fewer DOS related articles?"

JS: "I can't use them at present. I do not want to invest in PC equipment."

PL: "I'm sure many of our readers share your point of view regarding PC's. I know I took a long time to begin using mine with any frequency at all. I still get quite frustrated with the inability to easily do on the PC some of the things I have been doing, that now seems effortless, on the Poly. Thanks for taking the time to send in your responses, and I wish you continued success."

---

### Announcements

#### IBM PC'S RUN APPLE MACINTOSH SOFTWARE TODAY

San Jose, CA - hydra Systems, Inc. announces the delivery of its ANDOR ONE™ product, a PC add-in board with associated software that allows IBM PC's and compatibles to run Apple Macintosh® software. In light of the recent discussions between IBM and Apple, Hydra views its technology as a major step towards fulfilling the stated desires of both IBM and Apple to allow their products to work together.

The ANDOR ONE is a hardware and software package which combines the two predominant

microcomputer standards into a single machine. The ANDOR ONE works with all PC's ranging from XT's through 486's. This revolutionary product is seen as an alternative to Microsoft Windows for many PC users who would like to upgrade their computer's capabilities.

Hydra Systems, Inc. is located in San Jose, California and can be reached at (408) 253-5800 OR BY FAX AT (408) 253-1311.

---

### File Management

Have you ever postponed the backup process for weeks at a time? By the time you get around to it there may be many files that have been deleted from the sub-directory and dozens of that have been updated.

What is a good practice for backing up disks? When it comes to floppies, the easiest way is just to IMAG the source onto the backup. But when you are backing up on smaller sized drives this doesn't always work.

The Poly IMAG command actually only copies the used portion of a disk. The earliest version, Exec/4D, copied 350 sectors no matter how much data was on the disk. This would result in copying the blank space past the end of the used data. Later versions of Exec speeded up the process by only copying the used portion of the disk. So, if you IMAG from a large capacity disk with a small amount of data to a small capacity disk, the IMAG command would do the job without complaining (if the source data would fit on the destination disk).

But during most operations, the amount of data which actually needs to be backed up is usually much less than the entire disk. Only things that have actually been changed need to be backed up. The Poly BACKUP program selectively backs-up on the basis of files that have been changed. How does BACKUP.GO know when a file has been changed? It looks at the New bit and only copies the file if the new bit has been set [For more information on the New bit, see File Directory Entries in *PolyLetter* 91/3].

Once BACKUP has successfully copied the source copy of the file onto the backup copy, it clears the New bit. In ideal cases only a few files have been changed, and BACKUP.GO will do the job much faster than IMAG; this is only true provided you backup regularly. Also,

when the source directory has more data than the backup disk, IMAG can't be used at all. There is a DOS program which works like BACKUP.GO called BAC.COM and is available on the *PC Magazine* utilities diskettes. But suppose you are like many of us and procrastinate backing up until there have been lots of changes made, including adding and deleting files.

#### *Backing up after lots of delay*

Let's see now. If it has been a long time since we backed up a directory, we may have added and deleted several files. We will need to find out which files we should delete from the backup disk--BACKUP.GO will automatically add new files to be backed up. But we may want to check out the files we have deleted from the working directory to see if we want to restore one or more from the backup disk. There will be several things that must be done to make sure the source directory and the backup directory match. We won't be able to just compare the directories because the files may be in different orders; also, the disk address are likely to be different. These differences would prevent using Verify.GO on the directories themselves. Also, Verify.GO wants a file name, so can't check a directory against a root directory.

We have to compare each directory, delete the removed files--the ones we do not wish to restore--from the backup directory, restore the ones we want to keep, and copy the ones that have been changed. What a chore. Being a good procrastinator myself, I have created some command files to make the job easier.

The first thing I do is create a file with the names of the files in the source directory. Then I create another file with the names of the files in the backup directory. Next, I compare the two files. If the files compare ok, then the source and backup directories (disks) match in terms of file names. This is the most desirable condition and one need only run BACKUP.GO to complete the backup process.

To create the file of names of the files in the master directory, I use the Print-to-a-file utility (Fil.PS), and slist.GO. I usually name the file CD# (Compare Directory #). For the following examples, suppose I want to compare directory <4<DT against it's backup disk in drive 2. I will use drive 5 as the working drive to put the temporary files.

The command sequence to create a sorted list of the files in the master directory is:

```
Pr File <5<CD#.TX
?DISA
slist <4<DT /P
Pr Null
```

"Pr File <5<CD#" opens file CD# on drive 5 as the printer capture file. Anything sent to the printer will go into this file instead.

"?DISA" sets disable mode so the sorted list will only print the file name and size. The question mark prevents aborting command file mode if the system is not already enabled. (I always run my Poly in ENABLEd mode.)

The line "slist <4<DT /P" evokes the slist program and tells it to read directory DT on drive 4. The "/P" tells the slist program to send its output to the printer instead of to the screen. I wrote the /P option into this program for exactly this purpose; slist.GO omits the header when the output is directed to the printer. This will ensure that the list going to the printer (and being captured by the Print-to-a-file utility) contains only file size and names.

Finally, "Pr Null" closes the output file (<5<CD#).

Now that I have captured a sorted list of files in the master directory, I need to do the same for the backup directory. Since I always put my backup disk in drive 2, I call the list of files for the backup directory "CD2". The command sequence for capturing this list is:

```
Pr File <5<CD2.TX
?DISA
slist 2 /P
Pr Null
```

After executing this sequence of commands I can compare the two files to see if they are the same. For this purpose I just want to know if the files are the same so I use the program Verify.GO. Verify.GO compares the two files and reports that the files match, or it aborts command file mode and reports "Verify error!". After all, we are simply *verifying* that the second file (copy) matches the first file (master).

It wasn't too long before I got tired of repeating this sequence. I decided to automate it by creating a submit file with replace-

able parameters. I replaced the master file name with parameter %1 and the backup file name with parameter %2. I also put some documentation into the file to help me remember how to use it. (Remember, I procrastinate backing up so might forget how the file works.) The final submit file looks like this:

```
?DE <5<CD#,<5<CD2
; RL<Submit c<comp %1 %2

; %1 is the source directory on the HD
; %2 is the backup directory (usually 2)

Pr File <5<CD#
?DISA
slist %1 /P
Pr File <5<CD2
slist %2 /P
Pr Null
EN
U<Verify <5<CD# <5<CD2
DE <5<CD#,<5<CD2
```

The first line deletes the old files if they are still there. The second line shows me the correct syntax for executing the submit file. I keep the Submit program itself in sub-directory RL and the submit file (comp.SB) in sub-directory c. This allows me to type the file c<comp.SB, and it shows me how to use it. Also, since I always have the screen editor (edit.GO) installed, I can use it to move up to the second line, delete the semicolon, change %1 and %2 to the proper directory names, and press return and execute the process directly.

If the files match, then the last line deleted these two work files. But if they do not match then Verify will abort command file mode, and the two work files will not be deleted. If they don't match, I'll want to look at them both and compare them directly to see what files aren't in both places.

I compare the two files by using the "blinker method". Astronomers search for planets and comets by taking two photographs of the heavens a few days apart. Both photos are placed in a device called a "blinker" which flips a mirror back and forth so that the image presented to the astronomer switches between photographs. When the photos are properly aligned in the blinker all the fixed stars will be seen in the same place as one "blinks" from one photo to the other and back. But a planet or a comet, which will have moved relative to the fixed stars in a few days will appear to

jump back and forth as one blinks from one photo to the other. Our eye-brain is very good at detecting something moving against a fixed background, and the blinker exploits this capability very well. I have discovered/invented a way to adapt this technique to manually comparing files.

When searching for a particular text string the Poly editor seems to jump full blown to the new location. I use this to my advantage by putting a particular text string at the beginning of one file and putting an exact copy of that string at the beginning of the other file. (Both files are loaded into the editor at the same time.) The string I use is "####" and a return. Then I use CTRL-F to find the string for the first time. Then I can "blink" back and forth between the first file and the second file by using CTRL-C to move down and CTRL-G to move back. [I have modified Edit.GO to include a backwards search using CTRL-G.] By hitting CTRL-C and CTRL-G in rapid succession, The screen will "blink" back and forth between the two files. Any differences appear to jump out at me. A file missing from one list will cause the file names below it to appear to jump up and down one line as you blink back and forth.

Most Poly users do not have my upgraded Edit.GO, but there is an easy work around. This particular effect of CTRL-G can be achieved using an escape definition. Define ESC,G as CTRL-B,CTRL-C with the key sequence: "ESC = G ^ B ^ C ESC". Just to make things easier with switching, it is a good idea to also define ESC C as CTRL-C with the key sequence "ESC = C ^ C ESC". This allows one to easily press ESC C ESC G ESC C ESC G, etc and see the blinking effect.

Ok, this is fine for the first screen full of data. What if that matches? Then we must move the markers and blink on the next screens. You guessed it... I've got macros (that's what escape definitions are called) for moving both markers forward one screen as well as backwards one screen. What's involved is going to the beginning of the file, finding the first marker and moving it down one page, and finding the second marker and moving it down one page also. Since both markers are advanced one page, the next blinking will be on the second page. Since one can not delete block markers from inside a macro it's simpler to copy the marked block and delete the old marked block, markers included. So, the macro

must find "#### (RETURN)", put block markers around it, move down one page (CTRL-N), copy the marker (ESC CTRL-C) and delete the old one (ESC SHIFT-DELETE). I define ESC=F for the forward movement of the pair of markers and ESC=B for the backward movement of the pair of markers. Here's the definitions as you would type them in (don't type RETURN where I start a new line; there isn't enough room to print the whole definition on one line, so I have broken it at functional break-points.

```
ESC=F
^B^F####^M^
^[^T^[^Q^[^S^T^N^[^C^[^_
^C
^[^T^[^Q^[^S^T^N^[^C^[^_
ESC
ESC=B
^B^F####^M^
^[^T^[^Q^[^S^T^P^[^C^[^_
^C
^[^T^[^Q^[^S^T^P^[^C^[^_
ESC
```

The first line of each definition goes to the beginning of the file (^B) and finds the marker string. The second line places block markers around the marker string and moves ahead (back) one page (^N or ^P), copies the marker string (^C), and deletes the old block (^[\_). The third line finds the next marker string and the fourth line moves it.

^T is the left arrow  
 ^Q is the up arrow  
 ^S is the right arrow

With these definitions saved to file WP<AS.ED I can automate the process of loading the files and inserting the marker strings. I do that with a command file called "Compare.CD" Compare.CD contains the following:

```
ED <5<CD# XXX
```

```
####
^####
^TAB<5<CD2
```

```
^<WP<AS
^F
```

Compare.CD uses Edit to read in the first file. It then inserts the first marker text to mark the beginning of file CD#. It then inserts the second marker text to mark the beginning of file CD2. Next it uses the ESC CTRL-I (an ac-

tual Tab) to open file CD2. The "A" is a CTRL-A and tells Edit to read in from CD2. The "+WP<AS" tells Edit to load the escape definitions from file WP<AS. Finally, the "+F" tells Edit to execute the ESC F macro to move the markers to the bottom of the first page in both files. We are now ready to manually blink using ESC C and ESC G. Note: To get an actual < in the file press ESC, left arrow, DELETE, and CTRL-U. To get an actual / or \ in the file press CTRL-F, CTRL-E or CTRL-A, ESC, CTRL-U, CTRL-U and then delete the unwanted extra cursor.

Sometimes I have let the backup process go so long, or I have been moving files between disks, and am no longer sure which copy of one or more files has been changed. This is especially a problem if one has been renaming files. Exec clears the New bit when a file is renamed. If one renames the file on both the master directory and on the backup disk, then one might not be sure that the file doesn't need backing up. For this purpose, I have devised another command file which verifies each file in the master directory against the file with the same name in the backup directory. To construct this file by hand, one would need to capture the directory list to a file, strip the file of all but the file names, insert the directory location in front of each file name, copy the file name and insert the backup directory location in front of it, and insert the program name Verify on the command line in front of the file. Finally one would have to save the file and then execute it as a command file. Whew... I know, I sure did this enough times before I hit on a way to automate the process using slist (DIR could have been used) and Edit.

My automated submit file does the job with a much more complicated set of commands. Remember, the final command file is a sequence of instructions for verifying each file individually. It begins by capturing the sorted directory listing using print-to-a-file. But first, it makes sure the output file has been deleted.

```
?DE <5<VE#
Pr File <5<VE#
slist %1 /P
Pr Null
```

So far, so good. Now we must EDIT VE# (Verify Everything on #); each line now contains the file size and name. That must be changed to Verify source copy. For example, if a line had

the following:

5 document.TX

it would have to be changed to:

u<Verify <#<document.TX <2<document.TX

When the number of files is quite large the output to the printer may have form-feed characters (␣) in it; I begin by stripping those out. To do this I use the ESC colon command.

←:␣␣^[ε← or ←:^F^L^[^D←

Then we must get back to the beginning of the file with CTRL-B (␣).

We don't want to compare sub-directories; they are bound to be different since the files will usually be at different disk addresses. We strip out directories in a manner not unlike the way we removed form-feeds. Only this time we must delete a whole line rather than only one character.

←:␣.DX  
^[␣← or ←:^F.DX^M^[^X←

We must also delete the file size and all spaces on the line. Since the last item on a line is the file name, and it is preceded by a space, we can strip all the junk by searching for a space and deleting all before it. But first, we must get back to the beginning of the file with a CTRL-B (␣).

←:␣ ^[␣← or ←:^F ^[^X←

Once this process is complete, the only thing left in the file will be the file name and extension on each line. The next thing we will have to do is copy this, insert source and copy directories in front of the two copies, and put the Verify command at the beginning of the line. We will have to do this for each line in the file. We can do this by searching for a carriage return, but to insure that the first line gets processed correctly, we need to put a carriage return at the beginning of the file and move back to the top before starting the next process. First we go to the top of the file with a CTRL-B (␣). Then we insert the carriage return and do an up arrow (␣)

␣  
␣

What we do next will be done for each file by using the ESC : routine. We will find a carriage return, insert the command and a space on the line, insert the source directory, insert a beginning block marker, move to the end of the name by finding the next carriage return (CTRL-C [␣], left arrow [␣]) and backing up, inserting the end block marker, copy the name, insert a space, insert the backup directory, copy the name again and delete block; and we are going to do this inside an ESC : command so that it will repeat until the last carriage return is found. Whew... But to speed things up we will take advantage of the START command. After the first file name is processed, the remaining file names can be processed by using ST on the command line instead of Verify, since the Verify.GO program will still be in memory.

←:␣  
^[ST %1<^[␣␣␣^[␣^[␣ %2<^[␣^[␣\_←

A little cleanup will be needed since the last carriage return has no file name on it. We delete the extra ST with a CTRL-X [␣], go back to the beginning of the file with a CTRL-B (␣), and replace the first ST with U<Verify, and while we're at it delete that extra carriage return we put in to the top of the file.

␣␣␣ST←␣␣U<Verify

Now, all we have to do is exit from Edit with an ESC CTRL-E [␣␣] and then execute the command file <5<VE#. Let's put this all together in the submit file ver.SB so you can see what it looks like from within Edit.

?DE <5<VE#  
Pr File <5<VE#  
slist %1 /P  
Pr Null  
ED <5<VE#

←:␣␣^[ε←␣←:␣.DX  
^[␣←␣←:␣ ^[␣←␣␣  
␣←:␣  
^[ST %1<^[␣␣␣^[␣^[␣ %2<^[␣^[␣\_←␣␣␣  
ST←␣␣U<Verify←␣  
<5<VE#

The last line starts the file which was just created. It will keep going as long as files match, but will abort when the first file that does not match is Verified. This will allow you to check out those files and either restart <5<VE# or edit it to start with the file

which failed. The same or similar results can be achieved in DOS using a few batch file commands.

### Doing it in DOS

All this can be done much more easily in DOS. DOS batch files are much more powerful than Poly command files, although many things can be worked around, as comparing the following with the above will show.

The first thing we wanted to do above was to find out what files were in the backup directory but not in the source directory. To do this we must log into the backup directory and then execute a batch file which I call ISNOTIN.BAT. The batch file turns ECHO off so that it can echo only the files which were not found. Here's the contents of ISNOTIN.BAT:

```
@echo off
for %%a in (*.*) do if not exist %1%%a echo %%a is not in %1.
```

Here is a hypothetical run:

```
ISNOTIN B:\PL\
NEWFILE.TXT is not in B:\PL\
NEWFILE.DOC is not in B:\PL\.
```

This takes care of finding out what files are not in the backup drive. What about verifying them? I have not yet found a program like Verify.GO which runs under DOS, but there is one something like Compare.GO which can be used. The DOS COMP.EXE program will do the job for us. The strategy is like the ISNOTIN.BAT file, only instead of ECHOing the file name, we will compare them, and, of course, we will want to do it for files that exist, rather than ones which do NOT exist. I have called that file IFINCOMP.BAT and here it is:

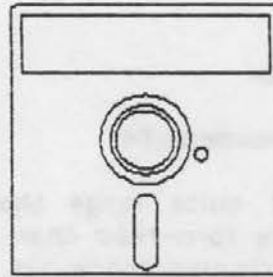
```
@echo off
for %%a in (*.*) do if exist %1%%a comp %%a %1%%a
```

Because of the way batch files work, we will be prompted at the end of each compare to see if we want to compare more files. We will answer NO each time. If there are no files missing from the backup directory, we could just use

```
COMP *.* B:\PL\*.*
```

This assumes that our backup directory is B:\PL. Pretty neat, huh?

### Above programs available



I have put all the Poly programs and command files mentioned above on a disk. If you would like to have the Poly software mentioned in this article send \$6.00, payable to Ralph Kenyon, to me at PolyLetter. You get AS.ED, comp.SB, Com-

pare.CD, Edit.GO & Efun.OV, edit.go Fil.PS, HELP.GO, slist.GO, Submit.GO, ver.SB, Verify.GO. Included on the disk are help files for the programs.

### Advertising

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Non-commercial ads by subscribers are free.

Spring vacation in the Berkshires for sale: One week time-share vacation townhouse at Oak and Spruce Resort, Lee, Mass. Sleeps 4, 1-1/2 baths, 19th week (starts on mothers' day). Dues paid to 1994. - Part of a swappable system. - Priced for quick sale at \$4000. Call 413-354-7750.

500 (count them) 4116 DRAM (16K to 64K upgrade chip) free for the asking: Limit 64 per customer, you pay shipping. Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA 93016-0385.

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all or swap for something interesting), Doug Schirripa, 716-624-370 (days) or 716-657-7437 (evening).

PolyMorphic System User Manual, System-88 User's Manual with Exec/96 addendum, & System-88 Operation Essentials On IBM disk. Al Levy, 516-293-8358.

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

#### DISKS - MODEMS - PROMS - SOFTWARE - SPELL

1. MAXELL diskettes: 5-1/4" hard sector - \$10 per box.
2. Used diskettes: 5-1/4" 10 hard sector - \$0.50 each.
3. Hayes Micromodem 100 (300 baud S-100 internal modem) - \$20.  
(If you don't have a modem this is a cheap way to go.)
4. HayesSys modem software (for the Micromodem 100) - \$10.
5. Abstract Systems Exec (Enhancements & bugs corrected) - \$30.
6. Abstract Systems Proms (Enhancements & bugs corrected) - \$35.
7. PolyGlot Library Volumes: \$6 each; 5 or more - \$5 each.
8. Hayes Smartmodem 1200B (IBM compatible internal) - \$30.  
Abstract Systems, etc., 191 White Oaks Road,  
Williamstown, MA 01267, Phone: (413) 458-3597  
(Send \$1.00 for a complete catalog--[free with any order].)  
(Make check or money order payable to Ralph Kenyon.)

### Help!

In this section I share with you the help system files I have built up over the last few years. (The entire system is included with Abstract Systems Exec.)

#### HELP BASIC DIM

HELP file for BASIC keyword "DIM".

DIM0 Start the array index with 0.

DIM1 (default) start index with 1

DIM declares a variable of type array.

DIM spec[,spec]

spec = numeric spec | string spec

numeric spec = numeric variable name(<exp>)

string spec = string variable name(<exp>:<exp1>)

exp = expression which evaluates to number of array elements

exp1 = expression which evaluates to size of array element

Example: DIM A\$(3:6),N(5),A\$(L:W)

#### HELP BASIC FUNCTION STR\$

HELP file for BASIC FUNCTION "STR\$"

"STR\$(expression)" returns the string with the specified numerical value.

Format specifications may be included, but they must follow the expression to be converted, and the # character must be included.

Example STR\$(4/3,%.4F2) => "1.33"

#### \$HELP Program Fil

HELP file for [A:]: program Fil.PS

Fil.PS is a custom Printer Driver which captures the output to the Printer and saves it in a file.

To open a capture file after Fil.PS has been installed, type:

\$Pr File <path>name.EX

If you forget the output file name, Fil.PS will ask for it.

To close an open capture file, select any defined printer.

#### \*\* CAUTION \*\*

Fil.PS does not know about other open output files. Make sure the capture file is on a drive with no open output files.

INSTALL Fil.PS using Setup.GO and the CUSTOM command. Give Fil.PS as the driver name. Use the standard dialogue.

#### \$HELP Program Verify

HELP file for [A:]: Program "Verify".

Verify.GO verifies that a copy of a file is an exact match. It compares the two files, byte by byte, that are specified on the command line. If there are any mismatches, Verify aborts command file mode and returns to Exec. If the files match, Verify reports that the "Files match."

Syntax: "Verify <n>path1<file-oneL.EX> <m>path2<file-twoL.EX>"

Example: "Verify <2>DATA.DT <3>BACKUP-DATA.DT"

#### \$HELP program slist

HELP file for system program "slist".

slist.GO is a program which alphabetically lists all files on a drive, including all files in all undeleted subdirectories.

Syntax: "slist [n]" (RETURN).

Example: "slist 2"

"slist" lists all files on the system resident drive.

"slist [n]" lists deleted files on drive n.

In ENABLE mode, deleted and system files are included.

REENTER reprints the list.

slist.GO was written by Ralph Kenyon of Abstract Systems.

#### \$HELP Program Submit

HELP file for [A:]: program Submit

Submit File(.SB) Pram1,Pram2,.....,Pram9

Submit allows executing a command file with parameters on the command line. Submit will feed the parameters to the command file whenever the percent character and a number are in the command file. For example, MOVE.SB, a submit file to move a file using system commands would have the following in it:

COPY %1 %2                    MOVE.SB would be activated by

DELETE %1                    "Submit MOVE <1>file <3>dfile"

Submit replaces "%1" with the first parameter supplied at

execution time. %0 is the name of the submit file itself. A "X" followed by something other than "0"-9" or another "X" will ask for input from the keyboard. "XX" passes one "X" on.

**Bit Bucket**

**FORMAT to WordPerfect**

The Poly WordMaster command [ne 6] causes FORMAT to skip to the head of the next page if there aren't 6 lines left on the current page. This is useful for insuring that tables, etc., stay together on the same page.

The same effect can be achieved in WordPerfect by using the Block Protect function. Place the cursor at the start of the area to be kept together. Then press Block (Alt-F4). Next, move the cursor to the end of the area you wish to remain together. Press Format (Shift-F8). WordPerfect will Prompt "Protect Block? No (Yes). Press "Y", and WordPerfect will insert the codes [Block Pro:On] and [Block Pro:Off] in the document around the block. These can be seen by pressing "Reveal Codes" (Alt-F3). When the document is printed WordPerfect will keep the block together on the same page.

**In This Issue**

Editorial . . . . .	1
Letters . . . . .	1
Survey-Interview - Jim Salinger. . . . .	2
Announcements . . . . .	3
File Management. . . . .	3
Doing it in DOS - File Management . . . . .	8
Disk of File Management Programs . . . . .	8
Advertisements . . . . .	8
HELP BASIC DIM . . . . .	9
HELP BASIC FUNCTION STR\$ . . . . .	9
HELP Program Fil . . . . .	9
HELP Program Verify . . . . .	9
HELP program slist . . . . .	9
HELP Program Submit . . . . .	9
Bit Bucket - FORMAT to WordPerfect . . . . .	10

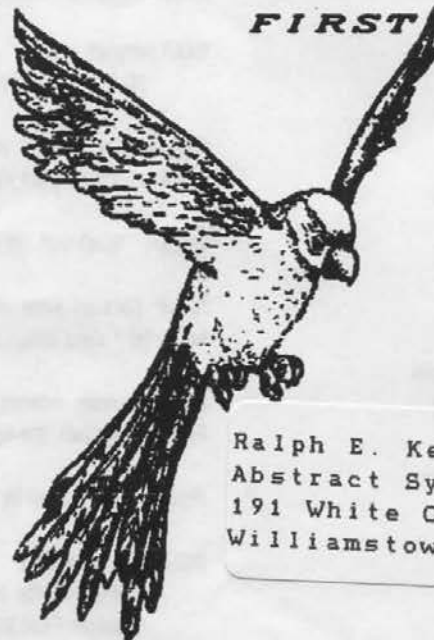
**Questions**

What questions would you like answered? Do you have answers? Write and tell us about things of interest to you. How do you use your Poly? Can you find and answer the questions asked in this issue? Send your answers and requests in.

**PolyLetter**  
 191 White Oaks Road  
 Williamstown, MA 01267  
 (413) 458-3597

Address Correction Requested

**FIRST CLASS MAIL**



Ralph E. Kenyon, Jr.                      EXP:99#9  
 Abstract Systems, etc.                      184  
 191 White Oaks Road  
 Williamstown, MA                              01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.  
 PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber adds are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Back volumes of PolyLetter (1980 through 1990) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (\$3.50,



# PolyLetter



PolyLetter 91/5

Page 1

SEP/OCT 1991

## Editorial

What do you do with an old Poly? Many of our number have stopped using their Poly's. In a few cases the machines have quit working, but mostly, people have gone on to more modern systems. One wonderful thing about the Poly, is that people are loath to throw it out. There is still some magic in that old hat, I mean, machine. I periodically send out mailings to expired Poly users (subscriptions, I mean, not subscribers), and I often get back replies like this:

"We still own our Poly. It's boxed up in our warehouse. Let us know if you are aware of anyone with a use for it.";

"Still have the Poly's, but one quit working... Now working in Epson I with 20 Meg HD and 386SX+150 Meg HD. No time to work with Poly."

"We have two Poly systems that we no longer use... If you know of anyone needing parts we would be interested in selling."; etc.

It breaks my heart to hear people say things like that, but I am heartened by the fact that they don't just throw it out. Others have said to me that they don't use it any more but can't bear to throw it out. One has it on display in his den where he can tell stories about it to visitors.

I have begun looking into the possibility that some school might use them in teaching electronics, but my inquiries have been to no avail. I thought about setting up a nursing home for old Poly's. Of course, if I ran a nursing home for old Poly's, I'd find ways to keep them young!

Another possibility is donating them. In

this regard there is the *East-West Educational Development Foundation*, "a charitable organization that places donated computer equipment in schools and educational institutions in the Soviet Union and Eastern Europe to assist them in their economic development as they move toward free enterprise and democratic societies." If you have an old computer which you would like to donate to this effort, you can contact the foundation at: One Exeter Plaza, 15th Floor, Boston, MA 02116. Telephone: (617) 542-1234; Fax: (617) 542-3333. Contact Al Hickey at (800) 442-1034 to arrange shipment. Donations are tax deductible.

Let's hear from our readers. What do you do with an old Poly? I make mine work harder than a PC! (Come to think of it, that's not hard to do.)

---

## Letters



Ralph, September 24, 1991

Just keep up the good work. -- Percy Roy, Edmonton, Alberta, Canada.

---

## *slist.GO and MSDOS 5.0*

When I want to get a sorted list of files on a Poly disk I use *slist.GO*. This program will construct an extended directory in memory with an index pointing to the entries in sorted order. It then checks to see if you have given it any parameters on the command line. The allowable parameters are leading characters of a file name, some wild cards, and some switches. For example, "*slist 2 a*" will list all files on drive 2 beginning with the letter "a".

I just checked the last issue of *PolyLetter* and realized that I published an earlier help file. The program *slist.GO* has been enhanced since that help file was written. You will find

the updated help file elsewhere in this issue. The enhancements include the options of listing System, Deleted, or New files or combinations thereof. `slist.GO` will also allow selectively list files with certain wild card parameters.

A question mark matches any letter, a star "\*" matches the remaining characters in the name or extension. For example, "`slist 2<DT *.TX`" lists all files with the extension .TX in sub-directory DT on drive 2. This includes all sub-directories on drive 2. "`slist ?a.`" lists all two letter file names ending in "a" on the system resident drive. "`slist ?a`" lists all files having the letter 'a' in the second position. "`slist 3 /N`" lists only New files on drive 3. "`slist 3 /ND` lists only Deleted New files on drive 3.

I use the `/N` option frequently to see how many files I need to backup. The `/D` option does the same thing as `DLIST`, except `slist` allows selectively listing files using wild cards to pick out certain ones. For example, "`slist P /ND`" will list deleted new files beginning with the letter P. Finally, the `/P` option will send the listing to the printer. The complete list of switches is `/DSNP`, and they may be in any order.

The new MS-DOS 5.0 allows one to accomplish a similar function with switches for the `DIR` command.

#### >MHELP DIR

Displays a list of files and subdirectories in a directory.

```
DIR [drive:][path][filename] [/P] [/W] [/A[:attributes]]
  [/O[:sortorder]] [/S] [/B] [/L]
```

```
[drive:][path][filename]
```

Specifies drive, directory, and/or files to list.

`/P` Pauses after each screenful of information.

`/W` Uses wide list format.

`/A` Displays files with specified attributes.

```
attributes  D Directories      R Read-only files
             H Hidden files    A Files ready for archiving
             S System files    - Prefix meaning "not"
```

`/O` List by files in sorted order.

```
sortorder  N By name (alphabetic)
            S By size (smallest first)
            E By extension (alphabetic)
            D By date & time (earliest first)
            G Group directories first
            - Prefix to reverse order
```

`/S` Displays files in specified directory and all subdirectories.

`/B` Uses bare format (no heading information or summary).

`/L` Uses lowercase.

Switches may be preset in the `DIRCMD` environment variable. Override preset switches by prefixing any switch with - (hyphen)--for example, `/-W`.

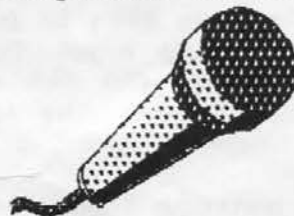
Poly:	MS-DOS 5.0:
<code>slist /n</code>	<code>DIR /ON /AA /S /P</code>
<code>slist DT ?A /N</code>	<code>DIR DT\?A*. * /AA /S /P</code>



I hope you all had a good halloween.

#### Survey-Interview: Al Levy

This is the third in a series of *Survey-interview's*. Only this time I called our respondent to "fill out" the survey interview. Our current interview is with Al Levy. Al is a (former?) Poly dealer who has moved into the PC world. He writes custom software systems that run under DOS for clients. Al is also the editor of *The Stack*, which is the newsletter for the Long Island Computer Association (LICA). The stack mailing list is about 2,000 strong. Al has been a frequent contributor to the *Poly-Letter*. Before getting involved with computers Al was a music arranger, pianist, conductor and teacher. Some of his credits include arrangements for the Jackie Gleason show and the Perry Como show, just to mention two.



PL: "Al, tell us about yourself. What do you do?"

AL: "I write and fill in

forms."

PL: "Don't be modest, Al. You've been around and done quite a bit."

AL: "Ok, I wrote 15 books on teaching music. I've written software for about 50 clients. I've been the past president of LICA."

PL: "What about current interests, this Midi thing?"

AL: "Midi stands for Music Industry Digital Interface. It's a standard interface for keyboards, synthesizers... It's build into the AMIGA and the MacIntosh, but not the IBM."

PL: "So, what do you need, a board for the PC?"

AL: "Yes. I have a Voyetra board and the software I'm using is Voyetra gold plus Finale. I just finished my 37th song on the computer."

PL: "Al, would you describe your Poly Systems for our readers?"

AL: "The system I use is an 8813 with three 5-1/4 inch drives; it also has two 8 inch drives. The Poly is connected to a Diablo printer. I have a dedicated cable connecting it to the DOS world with a switch box going every which way. Last year I thought the Diablo was dying, so I went out and bought a brand new Panasonic KXP-1191. I connected it to the Poly and was up and running in minutes. Try that in the DOS world. In other words, buy a printer that's not listed for your software on a DOS machine and see how quickly you have it running for each application."

PL: "What do you use the Poly for mostly?"

AL: "It's used mostly for business accounting and for the LICA mailing list. I used to run a mail order auction for out-of-print records. The entire auction was run on the Poly. I still look things up. If I want to look up a tape, see who's on it, etc., I grab a disk and throw it on the poly. Did that just yesterday. I've also copied all the important disks via the Poly Emulator to the IBM."

PL: "What other computers do you have?"

AL: "I have a host of DOS computers. There's an 8088 with a VGA display, an 8088 and a 286 (AT clone) both with monochrome displays. I also have both a 286 and a 386 with VGA moni-

tors. And there's the laptop. I use a mouse regularly."

PL: "You're actually a dealer for PC hardware too, aren't you?"

AL: "Hardware and software, right. Anybody in the Poly World who is looking for IBM compatible hardware can call and expect a nice discount by mentioning that they subscribe to PolyLetter."

PL: "Do you want to see more or fewer DOS related articles?"

AL: "I'd like to see more things about the Poly, unless you can draw a parallel. There's too much written about DOS already."

PL: "PolyLetter is regularly featuring articles about converting between the Poly and PC world. What kinds of Poly articles would you like to see in *PolyLetter*?"

AL: "Articles on using BASIC, what's available, articles on SYSTEM.SY, and about software and utilities in general."

PL: "What kinds of things would you like to see us print?"

AL: "What I'd most like to know is where I can get a Poly hard disk host adapter card."

PL: "Any concluding thoughts, Al?"

AL: "As long as the Poly runs, keep it going!"

PL: "Ok! Thanks, Al."

---

### **Converting ON ERROR & RESET**

BASIC C03, as well as later versions, provides the ability for the programmer to trap errors. The ON ERROR statement prevents BASIC programs from stopping when an error occurs. The way Poly BASIC works is that the last ON ERROR statement executed has control. It does not matter where the statement is in the program; it just has to have been executed. The syntax is:

```
ON ERROR <statement>
ON ERROR: <statement>
ON ERROR THEN <statement>
```

When an ON ERROR statement is executed

during normal program execution the <statement> following it is not executed. Executing an ON ERROR statement is like adding a "floating" GOTO statement to the program. The "floating" GOTO says GOTO <statement> from any point where an error occurs. During execution of ON ERROR A pointer internal to BASIC is set to tell BASIC where to go when an error occurs. The <statement> portion of the ON ERROR <statement> will only be executed if an error occurs. When an error does occur, instead of halting the program and reporting the error, BASIC transfers control to the <statement>. This feature allows programmers to "trap" errors and provide routines that allow the program to continue running.

According to the Exec/96 Addendum:

The first form causes the program control flow to be transferred to the <statement> immediately following the ON ERROR. Note that the BASIC control stack which keeps track of GOSUB and FOR/NEXT nesting is preserved.

The "colon" form also causes the program control flow to be transferred to the <statement> immediately following the ON ERROR. However, unlike the first form the BASIC control stack is reset. The "colon" form is useful for preventing stack overflows which often occur when using the first form.

The "THEN" form builds a "GOSUB" type environment when an error occurs and then transfers program control to the <statement> immediately following the ON ERROR. When the error handling routine executes a RETURN statement, program execution will resume after the statement which caused the error.

Error trapping can be turned off with a RESET statement. Here is a simple example of the first form which traps for a non numerical input error and which turns off the error trapping right after it is no longer needed. In general it is a good idea to turn error trapping off as soon as possible. Such a practice allows for more precise handling of errors. Don't you hate the programs that dumps you back to Exec (or DOS), without telling you what happened, when an error occurs? The following

program fragment takes control when an input error occurs.

```
1000 ON ERROR PRINT "Please try again."
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is",1/X1, "."
1030 RESET
```

Statement 1000 traps the error and transfers control to the statement which immediately follows the ON ERROR. In this case it is the PRINT statement. After the PRINT statement executes, control falls through to the INPUT statement again. There are two statements between the ON ERROR statement and the RESET statement, and the error could have occurred in either statement 1010 or 1020. The ON ERROR statement remains active until the RESET statement is executed.

The above example is not as simple as it might be because I wish to discuss converting ON ERROR statements to PC BASICs, and the GWBASIC INPUT statement normally does not generate an error. If you give a non-numeric answer it says "Redo from start?" and does not pass the error back for the ON ERROR statement to process--very annoying. The Poly fragment above could cause an error in the INPUT statement or from dividing by zero if X1 is zero. When this fragment is translated, only the division by zero error will be trapped by the ON ERROR statement. The INPUT statement will do its own internal trapping.

The second form--ON ERROR:--might be used in a loop or in a subroutine when the error trapping routine takes control outside the loop. Since the stack is reset, such an error trapping routine must take control outside ALL such loops or subroutines.

```
100 PRINT "Now we must input 6 numbers."
110 FOR I=1 TO 6
120 PRINT "Input the",I,"th number",
130 ON ERROR: GOTO 100
140 INPUT X(I)
150 RESET
160 NEXT
```

Since line 100 is outside the loop, repeating the error is likely to cause the stack to overflow and bomb the program, if not the system, were the first form used. The "colon" form resets the stack and prevents the overflow.

The "THEN" form would be useful for set-

ting a default value when the user simply hit return.

```
100 PRINT "Now we must input 6 numbers."
110 FOR I=1 TO 6
120 PRINT "Input the",I,"th number",
130 ON ERROR THEN 200
140 INPUT X(I)
150 RESET
160 NEXT
...
200 X(I)=-1 \RETURN
210 REM set bad input to minus 1.
```

There is a lot of power here. Much more complicated processing could be written.

#### ON ERROR in PC BASICS

A similar feature is available in GWBASIC on the PC.

The GWBASIC syntax is:

```
ON ERROR GOTO
```

Unfortunately, there are enough differences that direct translation isn't always possible. Like the Poly, the last ON ERROR statement executed remains in control. But unlike the Poly there is no RESET statement to turn ON ERROR off. And GWBASIC has a RESUME statement. The RESUME statement has 3 basic forms:

```
RESUME or RESUME 0
RESUME NEXT
RESUME <line number>
```

GWBASIC uses "ON ERROR GOTO 0" to turn off error trapping. However, if an error has occurred, the ON ERROR GOTO 0 does not work like Poly's RESET statement. If an error has occurred and no RESUME statement has been executed, ON ERROR GOTO 0 will stop the program and display the error almost as if there had been no ON ERROR GOTO <line number> executed.

RESUME or RESUME 0 works by sending control back to the statement that caused the error. The Poly program segment:

```
1000 ON ERROR PRINT "Please try again."
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is",1/X1, "."
1030 RESET
```

translates using the RESUME statement as follows:

```
1000 ON ERROR GOTO 2000
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is";1/X1; "."
1030 ON ERROR GOTO 0
...
2000 PRINT "Please try again."
2010 RESUME
```

Note: the commas in the Poly PRINT statement translate to semicolons in the PC PRINT statement. The way numbers print in GWBASIC is also annoyingly different, but I will discuss that in more detail at another time. Another difference is that line 1010 will not generate a trappable error. The only error this program fragment might cause is the division by zero error when X1 is zero.

GWBASIC has no equivalent for Poly's "colon" form of the ON ERROR statement. A program which uses an ON ERROR: statement can probably be rewritten with local error handling routines and loop or gosub flag variables to allow exiting from the routines when an error is detected, but that is beyond the scope of the present explanation.

Poly's ON ERROR together with a GOTO translates almost exactly to GWBASIC's ON ERROR GOTO <line number>. Here is a Poly program fragment using ON ERROR GOTO <line number>.

```
1000 ON ERROR GOTO 2000
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is";1/X1; "."
1030 RESET
...
2000 RESET
2010 PRINT "Fatal error in line 1010 or
1020."
2020 STOP
```

Line 2000 clears the error trapping on the Poly. On the PC the error flag must be reset before error trapping can be turned off. To reset the error flag for the purpose of executing an ON ERROR GOTO 0 one needs the RESUME <line number> statement.

GWBASIC's ON ERROR GOTO <line number> together with RESUME <line number> works like Poly's ON ERROR together with a GOTO.

PC translation:

```

1000 ON ERROR GOTO 2000
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is";1/X1;"."
1030 ON ERROR GOTO 0
....
2000 RESUME 2005
2005 ON ERROR GOTO 0
2010 PRINT "Fatal error in line 1020."
2020 STOP

```

The significant difference here is the insertion of the RESUME 2005, which turns off the error flag and allows the ON ERROR GOTO 0 to turn off error trapping. If the RESUME statement were not there, LINE 2005 would stop the program and report the error just as if there had been no error trapping at all.

Poly's ON ERROR THEN is normally followed by <line number> and may be understood as an implied indirect GOSUB. It is as if the statement which caused the error were actually a GOSUB <line number> in the statement which caused the error. When a RETURN is executed, control is transferred to the statement following the one in which the error occurred. This functions very similar to the GWBASIC RESUME NEXT statement.

Poly BASIC:

```

1000 ON ERROR THEN 2000
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is";1/X1;"."
1030 RESET
....
2000 X1=-1
2010 RETURN

```

PC BASIC

```

1000 ON ERROR GOTO 2000
1010 INPUT "NUMBER please? ",X1
1020 PRINT "The reciprocal is";1/X1;"."
1030 ON ERROR GOTO 0
....
2000 X1=-1
2010 RESUME NEXT

```

This covers the basics (get it?) of translating ON ERROR statements from Poly BASIC to PC BASICS (GWBASIC, etc.). Because of the flexibility of both error trapping routines a great number of variations is possible. I believe the Poly has many more possibilities since ON ERROR <statement> is not limited to ON ERROR [GOTO <line number>]. I would be happy to offer advice or assistance for any

possibilities our readers pose. Feel free to contact me if any of these translations poses problems I have not covered yet. [1-413-458-3597 till 11pm EST].

### BugNote - PC BASIC

I recently converted a Telephone Bill auditing program from PolyMorphic Systems BASIC version C04 to the QBasic I received with my recently acquired MS-DOS 5.0. I had some difficulties with some simple IF statements, so converted the program first to the GW-BASIC I purchased from TATUNG. That version came on a MS-DOS 3.2 disk. I had the program running correctly under GW-BASIC and ran it under QBasic. It failed in a very subtle way. I had built tax rounding error tests into the audit program and QBasic was not branching on IF Z=0 THEN when Z should have been zero. Subsequent research showed that the problem was present in GW-BASIC as well, although it did not appear often enough to make the audit program run incorrectly. I wrote a small test program which shows the error clearly. It seems that GW-BASIC and QBasic can't add and subtract correctly! I have reduced the error to what seems to be its simplest form. The following documents a rather severe error in GWBASIC from MSDOS 3.2 and QBASIC from MSDOS 5.0.

```

gwbasic
REM .26 + .04 - .3 = 0
print .26 + .04 - .3
-2.980232E-08 <--- This should be 0 !!
Ok
system
qbasic
print .26 + .04 - .3
-2.235174E-08 <--- This should be 0 !!

```

Here is a test program. Inserting various values will cause Z to be about 1E-08.

```

10 R0 = 3: F9 = 6.6: F8 = 1.25: F7 = .85:
20 B1 = F9 + F8 + F7
30 INPUT "TEST VALUE", BILL
40 T7 = BILL
50 COST = BILL + B1
60 TAX = INT(R0 * COST + .5) / 100: TOTAL = COST + TAX
70 PRINT "Base"; B1; "Bill"; BILL; "Tax"; TAX;
80 PRINT "Total"; TOTAL; INT(TOTAL * 100) / 100
90 Z = INT(R0 * T7 + .5) / 100 + INT(R0 * (F9 + F8 + F7) + .5) / 100 - TAX
90 IF Z = 0 THEN 110
100 PRINT "Z="; Z :REM Z should be exactly +/- .01 or 0 only.
110 GOTO 30

```

When the above program runs, Z is never zero. The serious nature of this bug can be seen when you understand that any

```
IF VAR=0 THEN <statement>
```

test will not always work correctly.

When is a BUG not a BUG?

After talking with MicroSoft I understand that this is not technically a bug on PC systems. To understand why, we must talk about how numbers are stored in the different systems. In Poly BASIC floating point numbers are stored in binary coded decimal (BCD) format. In PC BASICs floating point numbers are stored in binary format. For example, on the Poly the number .5625 is stored internally as a 56 25 00 00 (4 bytes for DIGITS 8, which is the default). There is also an exponent to say where to put the decimal point. On the PC, the number would be stored as 90 .. 00 (16 bytes) with an exponent to say where to put the decimal point when the number is converted back to decimal. I chose the number .5625 because it is  $9/16^{th}$ , and its binary fractional representation is exactly .10010000... which is .90 ... hexadecimal.

Any BASIC which uses BCD storage format will have (potentially) zero error when it comes to addition and subtraction of numbers with the same precision and within the storage limitation. But PC BASICs use the IEEE standard for floating point representation for personal computers. That standard calls for binary representation. In BCD representation .3 is just 30 ..., but in binary representation .3 is .01001100110011001100... binary or .4CCCC... hexadecimal. When the allotted 16 bytes (32 bytes for double precision) is used up, there is the matter of the leftover part which gets thrown away. Numbers which can be represented as sums of exact powers of  $1/16$  (up to  $1/16^{128}$ ) come out correctly. Anything with a bigger exponent comes out 0.

<u>QWBASIC</u>	<u>Poly BASIC</u>
print (1/2)^128	>PRINT .1^64
2.938736E-39	1E-64
Ok	>
print (1/2)^129	>PRINT .1^65
0	0
Ok	>

For us Poly users, who are used to the exact results obtainable with from 6 to 26 dig-

its of BCD precision, this problem is a definite bug. Fortunately, there is a way around it.

#### PC BASIC Bug Work-around

There is a work-around. Instead of testing for VAR=0, one test for ABS(VAR)<.000001. Convert any IF VAR=0 THEN statement to the above form. I have not examined the impact of this bug on <0 and >0 tests. Caution is urged. A similar work-around may be needed.

#### *Why Can't DOS ...*

IMAG between different sized disks?

Poly can IMAG between disks of different sizes. She can always IMAG from a small disk to a larger disk. She can sometimes IMAG from a larger disk to a smaller disk. How come the PC can't do the same? Why can't I use DISKCOPY to copy from 370K drives to 720K or bigger drives on the PC?

Poly's IMAG procedure works fine when disks are organized sequentially, as Poly disks are. But it can't work for DOS. DOS organizes disks as random access devices and there's no simple way to tell what part of the disk is unused. When Poly IMAGES disks Dfn1 looks at a particular spot in the root directory to see how much data is on the used portion of the disk. If that much data will fit on the destination disk, Dfn1 just copies the used portion of the source disk onto the same amount of space on the destination disk, including deleted files.

Since DOS treats disks as random access devices, it must keep track of whether each block is used or free. The Poly directory has a single pointer to the next free sector. Because Poly disks are organized sequentially, any writing is done to the next available sector and is done sequentially. Because DOS disks are organized randomly, DOS must have a pointer to each block which could be used for file storage. The list of pointers "maps" the disk usage; and is called the "File Allocation Table" (FAT for short). Rumor has it that the FAT is so important that DOS keeps 2 copies on each disk. There may be another reason for having two copies, but I don't know it. Anyway, when DOS DISKCOPIs a disk it copies the FAT, boot sector, root directory, and all disk sectors to the destination drive. A bigger disk would have to have a larger FAT, so Poly's IMAG strategy would leave portions of the larger

disk unmapped by a FAT and hence unusable. The large and the small of it is it just don't work for DOS.

There is a way to achieve the effect of a Poly PACK and IMAG or DIRCOPY on DOS disks, provided there is enough room on the destination disk. On early versions of DOS you would need to COPY \*.\* , but a much faster way for later versions of dos is XCOPY \*.\* . If the destination disk is blank, this strategy will also put fragmented files back together. What are "fragmented files", you ask? That's another story. I'll get to it in a future issue.

---

### Announcement - TICKLER.BAS

Abstract Systems, etc., announces the release of TICKLER.BAS into the DOS shareware marketplace. TICKLER.BAS is a simple program to remind the user of up-coming events as well as to follow-up on past due events. It reads a data file, which you update with a simple ascii text editor, and announces any events past due or scheduled within the next week.

I use TICKLER.BAS to keep track of all kinds of events--birthdays, holidays, conferences, mortgage due dates, etc. Each day the program lets me know about things which will be coming due during the next week; it also reminds me of events I haven't cleared.

TICKLER.BAS was originally written to run under PolyMorphic BASIC C04, but has been converted to run under DOS BASICs. The Poly version, TICKLER.BS was written up and the initial conversion to DOS BASIC was discussed in *PolyLetter* 88/5. Since then I have continued to enhance both versions of the program. The DOS version now generates its own sample data file. The shareware license fee is only \$10. Contact Ralph Kenyon at 1-413-458-3597.

---

### Advertising

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Non-commercial adds by subscribers are free.

---

Spring vacation in the Berkshires for sale: One week time-share vacation townhouse at Oak and Spruce Resort, Lee, Mass. Sleeps 4, 1-1/2

baths, 19th week (starts on mothers' day). Dues paid to 1994. - Part of a swappable system. - Priced for quick sale at \$4000. Call 413-354-7750.

---

500 (count them) 4116 DRAM (16K to 64K upgrade chip) free for the asking: Limit 64 per customer, you pay shipping. Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

---

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA 93016-0385.

---

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all or swap for something interesting), Doug Schirripa, 716-624-370 (days) or 716-657-7437 (evening).

---

PolyMorphic System User Manual, System-88 User's Manual with Exec/96 addendum, & System-88 Operation Essentials On IBM disk. Al Levy, 516-293-8358.

---

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

---

### DISKS - MODEMS - PROMS - SOFTWARE - SPELL

1. MAXELL diskettes: 5-1/4" hard sector - \$10 per box.
2. Used diskettes: 5-1/4" 10 hard sector - \$0.50 each.
3. Hayes Micromodem 100 (300 baud S-100 internal modem) - \$20.  
(If you don't have a modem this is a cheap way to go.)
4. HayesSys modem software (for the Micromodem 100) - \$10.
5. Abstract Systems Exec (Enhancements & bugs corrected) - \$30.
6. Abstract Systems Proms (Enhancements & bugs corrected) - \$35.
7. PolyGlot Library \$6 each volume; 5 or more: \$5 each; ALL: \$99
8. Hayes Smartmodem 1200B (IBM compatible internal) - \$30.

Abstract Systems, etc., 191 White Oaks Road,  
Williamstown, MA 01267, Phone: (413) 458-3597

(Send \$1.00 for a complete catalog--[free with any order].)

(Make check or money order payable to Ralph Kenyon.)

---

### Help!

In this section I share with you the help system files I have built up over the last few years. (The entire system is included with Abstract Systems Exec.)

◆HELP program slist



HELP file for system program "slist".

slist.GO is a program which alphabetically lists all files on a drive, including all files in all undeleted subdirectories. By specifying a path and directory, slist will only list files in that directory and its descendants.

Syntax: "slist [d<path>] [afn] [/NSDP]" (RETURN)

N - New files, S - System files, D - Deleted files  
 ND - Deleted New files. No parameter lists all files.  
 P - Send output to the selected print device.  
 <afn> - ambiguous file name. May include '\*' and '?'.

Example: "slist 2<DT> \*.TX /NP"

"slist" lists all files on the system resident drive.  
 "slist ?.\*" lists all one letter file names.  
 "slist 2 /DP " lists deleted files on drive 2 to the printer  
 (Note: Deleted New files will not be included.)

In ENABLE mode, deleted and system files are included.  
 REENTER reprints the list. REENTER may have new parameters.  
 "slist <4<TX /N" lists new files in directory <4<TX.  
 "REENTER W /N" lists new files (in directory <4<TX) beginning with the letter W.

slist.GO was written by Ralph Kenyon of Abstract Systems.

### Product Review: PKLite

I 'd like to bring your attention to the announcement by PKWare which appeared in *PolyLetter* 91/3 on page 2. I purchased PKLITE when I received the announcement, and it has saved me a LOT of disk space. This is one of those DOS programs whose performance or benefit lives up to the standard we Poly users have become accustomed to. We Poly users are accustomed to programs which are optimized for size--we might be called "byte misers". We are understandably annoyed by the size required for most DOS programs. PKLite, which compresses executable programs, brings much needed relief. For example, compressing WP.EXE:

pklite wp.exe  
 PKLITE (tm) Executable File Compressor Version 1.12 6-15-91  
 Copyright 1990-1991 PKWARE Inc. All Rights Reserved. Patent Pending

File: WP.EXE may contain overlays. Compress (y/n)? Y

Compressing: WP.EXE  
 Original Size: 228352 Compressed Size: 172199  
 Ratio: 24.6

The term Ratio is misleading. It is actually the savings in percent. The true disk space savings is determined by computing the number of whole blocks saved. For WordPerfect 5.1 that is:

Original Size: 228352 - 112 blocks.  
 Compressed Size: 172199 - 85 blocks.  
 Savings: 27 blocks = 54K (24%).

Some smaller executable files will compress with a smaller byte size, but won't actually realize a savings because the smaller byte size still requires the same number of blocks. 4000 bytes could be compressed by nearly half to 2050 but both require 2 blocks. I went wild compressing everything I could on my 30 Meg. HD. I must have gained 4 Meg. and was impressed. A couple of programs couldn't be compressed--LIST.COM, which must be able to modify itself during cloning & QMODEM, (I think).

Here's PKLite's help screen.

PKLITE (tm) Executable File Compressor Version 1.12 6-15-91  
 Copyright 1990-1991 PKWARE Inc. All Rights Reserved. Patent Pending

Usage: PKLITE [options] [d:|/path]Infile [(d:|/path)Outfile]

Options are:

- a = always compress files with overlays and optimize relocations
- b = make backup .BAK file of original
- e = make compressed file unextractable (\* commercial version only \*)
- l = display software license screen
- n = never compress files with overlays or optimize relocations
- o = overwrite output file if it exists
- r = remove overlay data
- u = update file time/date to current time/date
- x = expand a compressed file

(\* See documentation and license screen for more information

If you find PKLITE easy and convenient to use, a registration of \$46.00 would be appreciated. Registration includes one free upgrade to the software and a printed manual. Please state the version of the software that you currently have. Send check or money order to:

PKWARE, Inc.  
 9025 N. Deerwood Drive  
 Brown Deer, WI 53223

I think it's well worth the investment. If you order it based upon this review, mention *PolyLetter*.

### Bit Bucket - Poly Prompt

Do you like the Poly's "\$" prompt? ("\$\$" in

enabled mode.) Want to see the familiar, as one wag called it, "money sign" prompt on DOS machines? You can have this PolyFlavor by putting the following line in your AUTOEXEC.BAT file:

PROMPT \$P \$\$

(The \$P adds the current path to the prompt.)

Now if you are really homesick you can create two command files (batch or .BAT file in PC parlance).

ENABLE.BAT (or EN.BAT)  
PROMPT \$P \$\$\$

DISABLE.BAT (or DISA.BAT)  
PROMPT \$P \$\$

Of course, these will only change the command line prompt.

**In This Issue**

Editorial . . . . . 1

East-West Educational Development Foundation . . . . .	1
Letters . . . . .	1
slist.GO and MSDOS 5.0 . . . . .	1
MSHELP DIR . . . . .	2
Survey-Interview: Al Levy . . . . .	2
Converting ON ERROR & RESET . . . . .	3
ON ERROR in PC BASICS . . . . .	5
BugNote - PC BASIC . . . . .	6
PC BASIC Bug Work-around . . . . .	7
Why Can't DOS ...	
IMAG between different sized disks? . . . . .	7
Announcement - TICKLER.BAS . . . . .	8
Advertising . . . . .	8
HELP program slist . . . . .	8
Product Review: PKLite . . . . .	9
PKLite's help screen . . . . .	9
Bit Bucket - Poly Prompt . . . . .	9

**Questions**

What questions would you like answered? Do you have answers? Write and tell us about things of interest to you. How do you use your Poly? Can you find and answer the questions asked in this issue? Send your answers and requests in.

**PolyLetter**  
191 White Oaks Road  
Williamstown, MA 01267  
(413) 458-3597

Address Correction Requested

**FIRST CLASS MAIL**




Ralph E. Kenyon, Jr.                    EXP:99#9  
Abstract Systems, etc.                    184  
191 White Oaks Road  
Williamstown, MA                    01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.

PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber adds are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Back volumes of PolyLetter (1980 through 1990) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 Vol. - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (US: \$3.50, Canada: \$4.00, Overseas: \$5.00).

# PolyLetter



## The System-88 Users Newsletter

PolyLetter 91/6

Page 1

NOV/DEC 1991

### Editorial

I have gathered the impression that some of our readers think that *PolyLetter* is just for Poly computer users. This is not true. *PolyLetter* is also for users of System-88, the Poly Operating System, whether it runs under an actual Poly computer or whether it runs under the Poly Emulator on DOS machines. Of course, hardware articles may be of less interest to readers who are running System-88 on DOS machines. But Mitchell Lippman runs System-88 on a DOS machine, and he has suggested that he is interested in hardware add-on articles as well as in software articles.

I can think of hardware articles I would like to solicit from our readers. How about an article which interfaces modern drives to the Poly? What kind of S-100 controller would be required in order to attach a drive which writes its data in PC format? How about the new IDE drives; can they be interfaced to a Poly by using some simple host adapter card? I, myself, have been tempted to write a low-level disk device driver program which would allow the Poly to read and write to soft-sector disks, albeit in single density format. I haven't yet checked out the Poly's single density controller to see if it can find the index hole.

Let's hear from our readers who run System-88 on DOS machines. What experiences have you had integrating your Poly programs to DOS files? Can Poly command files running under the emulator control DOS programs? Can DOS .BAT files control Poly programs? What tricks and problems have you discovered while trying to make these two, admittedly very different, systems cooperate with each other?

Since I am still using my Poly Machine for a lot of stuff, most of my communicating is done over the serial port. I use Procomm on the PC and Bob Bybee's SM.GO on the Poly to transfer files back and forth.

Incidentally, anyone who is running System-88 on DOS machines under the Poly Emulator should consider getting and using Abstract Systems Exec/(A)S). It is available on DOS disk format as well as under Poly disk format. (See Abstract Systems' add on page 7.)

### Letters

Dear Ralph,

November 17, 1991

Thanks for yet another *PolyLetter*. And in answer to your question, no, there have been no more upgrades to the Poly Emulator program. Unfortunately, I don't expect to produce any, given the market size for something like that.



Regarding your BugNote article on floating-point numbers in PC BASIC: Your explanation of the problem is absolutely right. But few people would refer to this as a bug. It's just a well-known artifact of floating-point arithmetic on nearly all machines, one which programmers have learned to deal with. Often it causes no problem at all, since you rarely need to test for exactly zero on a floating-point number. You usually test for exact values on ordinal things like byte counts, for/next loops, and so on, where integer variables are used. If you use integer variables in PC BASIC or other PC languages, this problem doesn't occur.

For those rare instances where you do need "exact" floating-point precision on a PC, such as in handling large dollar amounts, there are third-party math libraries available just for that purpose. And they do their math operations in -- guess what? -- old Poly-style BCD arithmetic. Most of the good ideas are old ideas.

PL readers may be interested to know the

status of former PL editor Frank Stearns. He's alive and well, and working as a free-lance technical writer. Frank has leapfrogged most of us ... instead of simply moving into the PC world, he has already made the jump to workstation computers. Frank does his desktop publishing on a Sun SparcStation, running the Unix operating system, using a software package called FrameMaker. Highly professional stuff.

Me, I'm also doing a bit of technical writing. I've got articles coming out in both the C Users Journal and Windows/DOS Developers Journal over the next few months. And in general, I'm keeping busy and having a good time. Best regards,-- Bob Bybee, Stone Mountain, GA.

[Unfortunately, there's no cents to using integer variables for money calculations. -- Ed.]

Dear PolyLetter, November 29, 1991

I would appreciate articles on recommendations for interesting software and add-on equipment.

I am renewing for old time's sake. Thanks for all the years of help, especially when my Poly was down. I never understood most of the really technical stuff in *PolyLetter* but most issues were interesting never the less. I bought my Poly in 1978 quite by chance after stepping into the first computer store in Atlanta out of curiosity. With the help of Ken Williams we developed programs helpful for my dental practice. Over the years I have improved on the originals by doing my own programming. A couple of years ago I transposed my programs to GWBASIC but continued to use the Poly for tasks not easily rewritten for the PC. That is until I purchased the remarkable Poly Emulating program. Now my Poly just gathers dust. Once again thanks for all the years of support. The whole Poly experience was an enjoyable education. -- Mitchell S. Lippman, Marietta, GA.

[I have been sending out letters to people on the mailing list who have not subscribed in a long time. One reply to those letters might interest you. -- Ed.]

Ralph, December 4, 1991

We built a four-axis machine tool control system around the PolyMorphic CPU (E90014E) and video board (D90011A). We sold these from 1978 thru 1985. They worked just fine; however,

machine tool requirements forced us onward and upward. We have quite a pile of non-functioning hardware we can probably donate, but would like to keep a couple of functioning systems going here to support those machines still in use. Do you know of anyone in southern California who can repair the CPU, Video and Audio cassette interface?. -- Bob Lieberman, Demott Electronics Co. 14707 Keswick Street, Van Nuys, CA 91405.

[I'll run your letter in *PolyLetter* and see if anyone responds. Your best bet may be to pick up a working used 8813 system. There are enough around for sale at nearly scrap prices. Some Poly users who have moved on to newer machines are willing to donate their Poly to new homes for just the price of shipping. I am starting a data-file with the names of anyone who has a used Poly for sale or donation for the time when the few of us still running them need to get spare parts.

But there are also people who wouldn't dream of parting with their Poly. -- See the next letter. -- Ed.]

Ralph Kenyon,

December 4, 1991

I loved my Poly. I put it together back in 1976-77. Of course, I still have it. But I put it under my work bench about the time I went to the Heathkit H89. Now, of course, I'm trying to find a place to put my XT since I want to put together a 386 (and save a bundle).

I get so many magazines and newsletters (I have the 1<sup>st</sup> Byte) I'm reluctant to get another. But if you send me a copy of *PolyLetter* I can decide better. No, I don't want to sell or give away my Poly. I think of myself as having my own museum (except nobody will look at it). It was interesting to hear from you. -- Victor L. Ransom, Tinton Falls, NJ.

---

### **PRINT statements in Poly and PC BASIC**

When converting PC BASIC programs to Poly BASIC (and in reverse for the perverse), one immediately notices that PC BASIC PRINT statement use a semicolon to signal printing at the next character. This semicolon translates directly to a comma in Poly BASIC. Unfortunately the comma used in PC BASIC does not translate directly into Poly BASIC. The comma is used as a TAB character; it signals PC BASIC to tab

over to the next tab collum, which is an even the Poly by substituting ",CHR\$(9)," in place of ",".

The way numbers print in GWBASIC is also just different enough to be annoying. Poly BASIC always puts a space in front of a number and never puts a space after the number. GWBASIC sometimes puts a space in front of a number and always puts a space after the number. The space in front of the number is omitted when the number requires a negative sign; in GWBASIC the space in front of a number is actually the default missing "+" sign. In Poly BASIC the space is inserted as a delimiter before any sign. Here are some examples.

PRINT "->",6,"." in Poly BASIC  
will print "-> 6."

PRINT "->";6; "." in GWBASIC  
will prints "-> 6 .".

PRINT "->",-6,"." in Poly BASIC  
will print "-> -6."

PRINT "->";-6; "." in GWBASIC  
will print "->-6 .".

Input	Poly output.	GWBASIC output
6	-> 6.	-> 6 .
-6	-> -6.	->-6 .

When a number appears at the end of a sentence and one wants the period properly placed right adjacent to the number, one must go through some shenanigans to get it perfect. The way I got things to work right was to convert the number to a string using the STR\$( ) function. I thought I'd convert the number to a string and then lop off the extra space, but it turns out that GWBASIC drops the space when it converts the number to a string. -- Annoyingly inconsistent isn't it?

PRINT "->":STR\$(6): "." in GWBASIC  
will print -> 6.

PRINT "->":STR\$(-6): "." in GWBASIC  
will print ->-6.

For in-line printing of numbers in Poly BASIC we must remember to put a space after each number. GWBASIC does it for us.

Poly: PRINT "There were ",N," bottles of beer on the wall."  
PC: PRINT "There were ";N;"bottles of beer on the wall."

Poly BASIC provides numeric formatting statements which are similar to those in FORTRAN. The "%" character signals a formatting command for the following numeric print elements in a PRINT statement. Three types of notation are supported -- integer, fixed point, and scientific notation. The format character "%" is followed by a number which specifies the total size of the field. Following that is a character which selects which type the field is to be. "I" specifies that the number is to be an integer; "F" that it is a fixed point format; and E that it is scientific or "engineering" notation. In the case of both fixed point and engineering formats, another number specifies the number of digits which follow the decimal point. (There are none in the case of an integer.) This may sound somewhat complicated, but it's actually quite flexible and powerful.

Type:	Integer	FLOATING	Engineering
Syntax:	%NI	%NFD	%NED
Example:	%3I	%5F2	%8E1

Poly also allows adding the characters "C" -- for adding commas, "\$" for prefixing the number with a dollar sign, and "Z" for dropping trailing zeros. The character "#" also signals that the print format specification is to become the default format for future print statements. When format specifications are used for STR\$ conversions, the "#" character is required. Also, the Z specification does not work correctly with the engineering notation.

Here are some examples.

PRINT "=>",&8I,6,"<="

will print "=> 6<="

PRINT "=>",&8F1,6,"<="

will print "=> 6.0<="

PRINT "=>",&8BF1,6000,"<="

will print "=> 6,000.0<="

PRINT "=>",&8F2,6,"<="

will print "=> \$6.00<="

PRINT "=>",&8ZF1,6,"<="

will print "=> 6. <="

PRINT "=>",&8E1,6,"<="

will print "=> 6.0E+00<="

PRINT "=>",&8BE1,6,"<="

will print "=> 6E+00<="

```
PRINT "=>,%C,7.5*10^6,"<="
    will print "=> 7,500,000<="
```

FORTRAN-like format statements do not work in GWBASIC. GWBASIC uses PRINT USING statements. A PRINT USING statement works more like COBOL than like FORTRAN. COBOL gives pictures of what the output should look like. In GWBASIC the "#" character is used to represent a digit in the picture of the output.

The above statements translate into PRINT USING statements only partially.

```
PRINT %8I,...
    translates to
        PRINT USING "#####";...
```

```
PRINT %8F1,...
    translates to
        PRINT USING "#####.#";...
```

```
PRINT %C8F1,...
    translates to
        PRINT USING "#####.##";...
```

```
PRINT %$8F2,...
    translates to
        PRINT USING "$$###.##";...
```

```
PRINT %Z8F1,...
    does not translate
```

```
PRINT %8E1,6,...
    translates to
        PRINT USING "#####^ ^ ^ ^"
```

```
PRINT %Z8E1,...
    does not translate
```

```
PRINT %C,...
    does not translate
```

The statements that do not translate may be approximated by first converting the variable to a string using the STR\$( ) function. The string obtained may then be manipulated into the desired format and printed as a string. Of course, getting the desired form may be more complicated than it's worth.

To get an output compatible with the %ZNFD format specification, one would need to do something like the following. Suppose X is the number which would be printed in a fixed point format N digits wide and with D decimal places. First we need to get rid of any extra digits there may be. We round off any extra

digits from the number using the integer function. Since we can't tell INT to use any number of digits, we must shift the decimal point first by the number of digits, D. That is accomplished by multiplying the number by  $10^D$ . To insure that the number is rounded properly, we must add .5. INT( ) would discard any fraction of a number larger than .5, and such a fraction must be rounded up. For example, INT(1.6) is 1 but INT(1.6+.5) is INT(2.1) and is 2 -- the correct answer. Once INT has rounded the number, we can then shift the decimal point back where it belongs. To do this we divide by  $10^D$ . The entire conversion can be accomplished in just one statement:

$$X = \text{INT}(X * 10^D + .5) / 10^D$$

We can make this conversion process easier by defining a function which rounds the result for us. Such a function would also need to specify the number of digits (the "D" in "%NFD"). We can use the above rounding formula directly to define the function.

$$\text{DEF FN ROUND}(X,D) = \text{INT}(X * 10^D + .5) / 10^D$$

Once this is done, we can put the number into a string.

$$X\$ = \text{STR}\$(\text{FN ROUND}(X,D))$$

But to know many extra spaces to put in the output string, we need to know how many digits actually precede the decimal point. We know from the string conversion that the length of the string will include an extra leading space, but how many actual digits precede the decimal point? We can find this out by using the relation between a number and its logarithm.

A base 10 logarithm has two parts, the mantissa and the exponent. The exponent is written in front of the decimal point and the mantissa follows. The exponent tells us how many digits to shift the decimal point. For any number X (bigger than zero) the exponent of the base 10 logarithm is just one less than the number of significant digits. If the number is negative, then it represents the number of times the decimal point must be shifted to the left. Since BASIC puts these zeros in, we only need to worry about a logarithm with an exponent which is positive.

Unfortunately, GWBASIC does not have base 10 logarithms; it only has natural logs.

Poly BASIC has both LOGT (base 10) and LOG (natural, or base e) functions built in. But we can get around this limit. It is possible to convert LOG functions to LOGT by dividing the result by the LOG of 10. The equation is:

$$\text{LOG}_{10}(X) = \text{LOG}_e(X) / \text{LOG}_e(10)$$

As long as we are at it, we might as well define a user defined function in GWBASIC which gives us LOGT directly. Such a function can be defined as follows:

```
DEF FN LOGT(X)=LOG(X)/LOG(10)
```

We can compute the number of significant decimal points of a number X by computing the FN LOGT(X), taking the exponent part of the result, and by adding one to it. The exponent part is just the part before the decimal point and the INT function is ideally suited for picking this out. Let's define a LDIGITS functions which will compute the result.

```
DEF FN LDIGITS(X)=INT(LOGT(X))+1
```

But, there are two problems with this. One problem is that the log of a negative number is not defined. To make this function work in the case of negative numbers, we must make sure X is positive. We can do this by using the absolute value function ABS. Here is the revised function.

```
DEF FN LDIGITS(X)=INT(LOGT(ABS(X)))+1
```

Unfortunately, this will cause an error when X is zero. I can't think of a way to test for this and give a different result in a single line function definition, so we will leave it as it is and leave the testing up to the user before the function is called.

What does this give us? It tells us how many leading digits to allow in the format specification. Since GWBASIC puts a space (or a minus sign) in front of a number, we can add 1 to this to tell us where the decimal point must go in the translation of the %ZNF format specification.

```
REM Condition X for proper rounding
X = ROUND(X,D)
REM Get it into a string
X$ = STR$(X)
REM compute the number of leading digits
REM & add 1 for the space
D1 = FN LDIGITS(X)+1
```

The second problem is that if the result is an integer, the STR\$ function will not put in a decimal point. We will have to do that manually. We can test for this condition by using the INT function again. If the number is a whole number, that is, if the integer part of the number is equal to the number, then we know that the STR\$ conversion is a whole number, and we must add the decimal point.

```
REM If necessary, add decimal point
IF X=INT(X) THEN X$=X$+."
```

We are still not ready to print the result; first we must create a print variable the right length. If N is the total length and D is the number of digits, then N-D-1 is the number of places in front of the decimal point. (The decimal point counts as one of the characters in the count N.) D1 is the actual number of digits (plus the space) in the string, so N-D-1-D1 is the number of spaces that must be added to the front of the string.

```
REM add missing leading spaces
IF N-D-1-D1>0 THEN
  X$=SPACE$(N-D-1-D1)+X$
```

X\$ will now contain the value just as the Poly PRINT statement would print it, but with the missing trailing spaces. We can correct for this by adding trailing spaces as necessary.

```
100 IF LEN X$<N THEN X$=X$+" " : GOTO 100
```

Since %ZNF does not work correctly on the Poly, there's no sense trying to translate it.

PRINT USING may be used for printing to files as well. The syntax is:

```
PRINT #N,USING <format expression>
```

But there is one Poly function GWBASIC has no equivalent for at all. Poly BASIC allows using format statements in string conversions. For example,

```
X$=STR$(2.54)
```

makes the same string as

```
X$=" 2.54".
```

But

```
X$=STR$(2.54,%#7F2)
```

makes the same string as

```
X$=" 2.54".
```

(Note: the "#" character is required in formatting commands used in the STR\$ function. I suspect it has something to do with calling the formatting routine without printing, but don't have access to the source code to check it out.) GWBASIC can't use the USING format statement in the STR\$ function. But there are tools to make this kind of formatting possible.

GWBASIC has LSET and RSET commands which position one string variable in another. LSET left justifies the input, while RSET right justifies the input. For example, in the above statement, we could create a string variable which is 7 spaces wide with the command

```
A$=SPACE$(7)
```

If we put our number into a string variable, then we can position it right justified in the A\$ string with the RSET command.

```
X$=STR$(2.54)
RSET A$=X$
```

These three commands give the same result as the Poly command:

```
A$=STR$(2.54,%#7F2)
```

Since GWBASIC won't allow multiple line user defined functions, we would need to define such a "function" as a subroutine.

```
REM DEF STR$(X,N,D)
REM result returned in STRING$
STRING$=SPACE(N)
RSET STRING$=STR$(FN ROUND(X,D))
RETURN
```

The above routine works fine when the number of digits in the output is exactly the number of digits desired by the format specification. But it doesn't work correctly when there would be a trailing zero.

Suppose the Poly STR\$ function returned a string with one or more zeros on the end. For example, STR\$(2.54,%#7F3) returns a string value of " 2.540". GWBASIC STR\$(2.54) returns a string value of " 2.54". We will need to test for this failure in order to get things exactly

right. How can we do that? Well, we can find out how long the resulting string is with the LEN function. But we will want to know if the number of digits after the decimal point is less than the number D in %NFD. Suppose our number is X. We can test how many decimal points will be in the rounded answer by throwing away the integer part of the number. Note that I said "in the rounded answer". We must first round the number using the rounding function. Then we can discard the integer part and test the result.

```
REM Round to the desired digits
TEMP = FN ROUND(X,D)
REM we need positive numbers
TEMP = ABS(TEMP)
REM Discard the integer part
TEMP = TEMP - INT(TEMP)
REM Convert to a string.
TEMP$ = STR$(TEMP)
REM Get length of string
TEMPL = LEN(TEMP$)
```

At this point we have a string, TEMP\$, which contains the non-integer part of X rounded to the correct number of digits (D) but with missing trailing zeros. There is the special case when the value is " 0". The length will be 2; any other value will be, at minimum, of the form ".d" and will have a length of 3 or greater. If we subtract 2 from the resulting length, we will get the actual number of decimal points in the string. But, if the size is exactly 2, we will need to add the decimal point manually.

Ok, now we are ready to proceed with the conversion.

```
REM Round to the desired digits
WORK=ROUND(X,D)
REM we need positive numbers
TEMP = ABS(WORK)
REM Discard the integer part
REM & convert to a string.
TEMP$ = STR$(TEMP - INT(TEMP))
REM Get length of digits
TEMPL = LEN(TEMP$)-2
REM Get our working string conversion.
WORK$=STR$(WORK)
REM If its an integer, add decimal point
IF LEN(TEMP$)<3 THEN WORK$=WORK$+"."
```

At this point TEMPL has the actual number of decimal points in the string WORK\$, and D is the number we want to have. Now we can proceed in two ways. Which way we go depends



upon whether we want to convert %NFD or %ZNFD. If we are converting %NFD, we will want to add missing 0's; if we are converting %ZNFD, we will want to add missing spaces. Suppose we make CHAR\$="0" in the former case and CHAR\$=" " in the latter case. We are now ready to add the missing characters.

```
99 CHAR$="0" :SEM %NFD
99 CHAR$=" " :REM %ZNFD

100 IF D<=TEMPL THEN 200
110 WORK$=WORK$+CHAR$ :REM Add one
120 TEMPL=TEMPL-1 :REM Count down
130 GOTO 100 : REM Go try again

200 REM
```

At this point, the decimal part of the string is filled out with extra 0's or spaces as necessary to put things in the right place. The rest is actually quite easy. As above, we define a variable with the proper length, and then we right set the working variable into place.

```
FINISHED$ = SPACE$(N)
RSET FINISHED$ = WORK$
```

All this was because GWBASIC does not allow formatting specifications in the STR\$ function. Bummer. Kind of makes one want to stick to the Poly, or at least to System-88 under the Emulator.

---

### Advertising

Commercial advertising rates are \$50 for a full page, \$25 for a half page, and \$15 for a quarter page. Anything smaller is \$3.00 per column inch. A column is 3-3/4 inches wide by 10 inches tall. A full page is 7-5/8 inches wide. Non-commercial ads by subscribers are free.

---

Mothers Day vacation in the Berkshires for sale: One week time-share vacation townhouse at Oak and Spruce Resort, Lee, Mass. Sleeps 4, 1-1/2 baths, 19th week (starts on mothers' day). Dues paid to 1994. - Part of a swappable system. - Reduced price - \$3,500. Call 413-354-7750.

---

Wanted to buy -- any and all Poly computers. 88, 8810, 8813, twin, 8824; documentation, software, keyboards, spare parts, etc. -- Call Charles Steinhauser - Phone: (404) 299-6123 after 7 pm. EST.

---

PolyMorphic 8813 needs home. Make offer. Conway Spitler, P. O. Box 385, Fillmore, CA 93016-0385.

---

Poly 8813, 3 drives, 64K, 2 printer ports, with a set of spare boards, 1 spare drive, 2 AJ-832 daisy-wheel letter quality printers, all documentation, etc. Any reasonable offer (must take all or swap for something interesting), Doug Schirripa, 716-624-370 (days) or 716-657-7437 (evening).

---

PolyMorphic System User Manual, System-88 User's Manual with Exec/96 addendum, & System-88 Operation Essentials On IBM disk. Al Levy, 516-293-8358.

---

FOR SALE: Poly 8810 box with power supply and mother board. \$50 plus shipping. Charles A. Thompson, 2909 Rosedale Avenue, Dallas, Texas 75205-1532, (214)-368-8223.

---

### DISKS - MODEMS - PROMS - SOFTWARE - SPELL

1. MAXELL diskettes: 5-1/4" hard sector - \$10 per box.
2. Used diskettes: 5-1/4" 10 hard sector - \$0.50 each.
3. Hayes Micromodem 100 (300 baud S-100 internal modem) - \$20. (If you don't have a modem this is a cheap way to go.)
4. HayesSys modem software (for the Micromodem 100) - \$10.
5. Abstract Systems Exec (Enhancements & bugs corrected) - \$30.
6. Abstract Systems Proms (Enhancements & bugs corrected) - \$35.
7. PolyGlot Library \$6 each volume; 5 or more: \$5 each; ALL: \$99
8. Hayes Smartmodem 1200B (IBM compatible internal) - \$30. Abstract Systems, etc., 191 White Oaks Road, Williamstown, MA 01267, Phone: (413) 458-3597 (Send \$1.00 for a complete catalog--(free with any order).) (Make check or money order payable to Ralph Kenyon.)

---

### Help!

In this section I share with you the help system files I have built up over the last few years. (The entire system is included with Abstract Systems Exec.)

#### \$HELP BASIC FORMAT

HELP file for BASIC PRINT statement formats. "%" specifies a numeric print format to follow. All numbers in the same print line following the format specification will be printed in the same format. If the formatted number will not fit, asterisks will be substituted. %21,100 ==> \*\*.

A format specification consists of the width of the field, the format type, and the number of decimal points to be used (if applicable)

Format types include:

I - Integer format Example I1 -3245645  
 F - Floating point format Example F2 -2343.45  
 E - Exponential format Example E2 -1.23-10

Note: when a format specification is used in the STR\$ function it follows the expression to be converted, and must include the default character "#". Example: STR\$(V,%#3I)

Additional formatting is accomplished with format characters.

Format characters include:

C - Place commas to the left of the decimal point.  
 \$ - Place a dollar sign to the left of the number.  
 Z - Eliminate trailing zeros.  
 @ - Set the new default parameters.

("##" resets the system default specifications.)

Finally, the TAB command is used to skip to the position specified by the expression in TAB(expression).

\$HELP BASIC FUNCTIONS

HELP file for BASIC FUNCTIONS

BASIC has the following regular intrinsic functions.

SQRT EXP LOG LOGT COM SIN TAN ABS INT SGN RND  
 TIME COSH SINH TANH ATAN ASIN FREE(0) MEM INP OUT  
 POKE PEEK LEN VAL STR\$ ASC CHR\$ LEFT\$ RIGHT\$ MID\$

to get HELP for one of these functions type  
 HELP BASIC FUNCTION <name>

\$HELP BASIC FUNCTION STR\$

HELP file for BASIC FUNCTION "STR\$"

"STR\$(expression)" returns the string with the specified numerical value.

Format specifications may be included, but they must follow the expression to be converted, and the @ character must be included. Example STR\$(4/3,%#4F2) => "1.33"

### Annual Index - 1991

8" Hard Disk - Priam - Source Ralph Kenyon 91/2/09  
 Announcement - Automate Plans Out of your Mind... 91/3/02  
 Announcement - BarZIP Charles Eglinton 91/1/01  
 Announcement - BUGS Insect control GardenTech 91/3/02  
 Announcement - Compress Executable PKWare 91/3/02  
 Announcement - Desert Storm on CDROMTime Magazine 91/3/02  
 Announcement - DOS 3-1/2" Disk Blazie Engineering 91/1/01  
 Announcement - Encyclopedia, windowsCompton 91/3/02  
 Announcement - Maintenance kit American Chemical 91/2/01  
 Announcement - PC based Ada compilerGEM Technologies 91/3/02  
 Announcement - PC runs MAC software Hydra Systems, Inc 91/4/03  
 Announcement - Pony Express Melisco Marketing 91/1/01

Announcement - TICKLER.BAS Abstract Systems 91/5/08  
 Annual Index - 1991 PolyLetter 91/6/08  
 Back in the USSR - 240V Power supplyRalph Kenyon 91/3/02  
 Bit Bucket - Addendum to Adventure Bob Bybee 91/2/10  
 Bit Bucket - Adventure Charles Steinhauser 91/1/10  
 Bit Bucket - Chomping at the Bit Ralph Kenyon 91/1/10  
 Bit Bucket - Format to WordPerfect Ralph Kenyon 91/4/10  
 Bit Bucket - HDTV PolyLetter 91/6/10  
 Bit Bucket - Loose Ring Error PolyLetter 91/6/10  
 Bit Bucket - Poly Prompt PolyLetter 91/5/09  
 BugNote - PC BASIC Ralph Kenyon 91/5/06  
 BugNote 20 -- Bug in Gfid and @ PolyLetter 91/1/09  
 BugNote 21.0 - Gfid: Disk Full errorAbstract Systems 91/3/09  
 Converting BASIC programs to GWBASICRalph Kenyon 91/3/04  
 Converting FW's to PC DOS BASIC Ralph Kenyon 91/1/04  
 Converting ON ERROR & RESET Ralph Kenyon 91/5/03  
 Direct Mail Marketing Association Ralph Kenyon 91/1/09  
 Drives Halt -- Stopping SD drives Ralph Kenyon 91/2/04  
 East-West Educational Development Foundation 91/5/01  
 Editorial Ralph Kenyon 91/1/01  
 Editorial - Ralph Kenyon 91/5/01  
 Editorial - For System-88 Users alsoPolyLetter 91/6/01  
 Editorial - Time flies Ralph Kenyon 91/4/01  
 Editorial - Transition to PC Ralph Kenyon 91/3/01  
 Editorial - Write Congress Ralph Kenyon 91/2/01  
 File Management - Backup Ralph Kenyon 91/4/03  
 HELP BASIC DIM Abstract Systems 91/4/09  
 HELP BASIC FORMAT Abstract Systems 91/6/07  
 HELP BASIC FUNCTION STR\$ Abstract Systems 91/4/09  
 HELP BASIC FUNCTION STR\$ Abstract Systems 91/6/08  
 HELP BASIC FUNCTION VAL Abstract Systems 91/3/10  
 HELP BASIC FUNCTIONS Abstract Systems 91/6/08  
 HELP COMMAND flip PolyLetter 91/1/09  
 HELP COMMAND fold PolyLetter 91/1/09  
 HELP COMMAND full PolyLetter 91/1/09  
 HELP Program Fil Abstract Systems 91/4/09  
 HELP Program slist Abstract Systems 91/4/09  
 HELP program slist Abstract Systems 91/5/08  
 HELP Program Submit Abstract Systems 91/4/09  
 HELP Program Verify Abstract Systems 91/4/09  
 HyperText under PC DOS Ralph Kenyon 91/1/05  
 Junk Mail - DMMA Ralph Kenyon 91/1/09  
 Letter Bob Bybee 91/6/01  
 Letter Mitchell S. Lippman 91/6/02  
 Letter - Commodity trading Secrets Jim Salinger 91/4/01  
 Letter - HD/18 Hard disk problems Jim Salinger 91/1/02  
 Letter - Need PC conversion info Gary Sterling 91/3/01  
 Letter - Poly Emulator Tom Bucy 91/1/01  
 Letter - Poly Emulator & HD/18 Bob Bybee 91/2/01  
 Letter - Poly HD & Commodity tradingJim Salinger 91/3/01  
 Letter - Weaning from Poly to PCDOS Earl Gilbreath 91/2/02  
 Letter - Why not compete with Poly Bob Bybee 91/1/02  
 MSHELP DIR PolyLetter 91/5/01  
 New Tricks - BACKUP.GO & REBUILD.GO Ralph Kenyon 91/3/03  
 Obituary - Constantin Pavloff Ralph Kenyon 91/2/03  
 ON ERROR in PC BASICS Ralph Kenyon 91/5/05  
 PC BASIC Bug Work-around Ralph Kenyon 91/5/07  
 PD disk with Edit, edit, Fil, etc PolyGlott Library 91/4/08

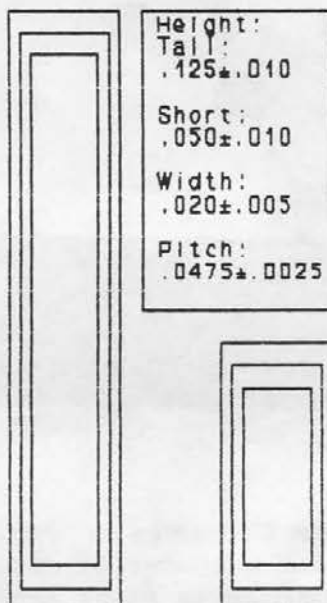
PKLite's help screen	PolyLetter	91/5/09
PolyLetter Back Issues Clearance	PolyLetter	91/2/02
Postal Barcodes - Part 1	Ralph Kenyon	91/6/08
PRINT statements in Poly & PC BASIC	Ralph Kenyon	91/6/02
Product review: PKLITE	PolyLetter	91/5/09
Review - DOS HELP! under PC DOS	Ralph Kenyon	91/1/06
Review - SuperMicro	Ralph Kenyon	91/1/06
Review - TechHelp under PC DOS	Ralph Kenyon	91/1/06
slist.GO and MSD 5.0	Ralph Kenyon	91/5/01
Some Observations	Charles Steinhauser	91/1/02
SPG - File Directory Entries	Ralph Kenyon	91/3/03
Survey - Interview with Bob Bybee	PolyLetter	91/3/09
Survey - Interview with Jim Salinger	PolyLetter	91/4/02
Survey form for readers.	PolyLetter	91/1/07
Survey-Interview - Al Levy	PolyLetter	91/5/02
System Programmers Guide - FDE's	Ralph Kenyon	91/3/03
Why Can't DOS IMAG between Different Sized Disks (rk)		91/5/07

### Postal Barcodes

Have you noticed that more and more mail has POSTNET barcodes? I heard that the Post Office was going to offer a discount for pre-barcode first class mail, so I decided to modify my label printing program to include barcodes. Then I found out that there is a volume requirement as well. I don't know what the minimum is, but it is probably well over anything I would mail at one time. At any rate, the exercise proved to be an interesting programming experience.

Printing barcodes on labels requires a printer which is capable of printing graphics. I made up the barcode by switching the printer into graphics, printing the barcode, and exiting graphics mode. It sounds simple in principle, but there are many details to consider.

I contacted the Post Office; they were very happy to send me a document giving the printing specifications for postnet barcodes. Postnet barcodes are made up of 5, 9, and 11 digit postal ZIP codes. I chose to implement 5 and 9 digit codes. Each barcode consists of a pair of tall framing bars between which the bars representing the digits are printed. Each



code also includes an extra checksum digit to allow for error correction. It's sort of like the parity bit in serial ports. The check digit is computed by adding up all the other digits and then figuring out what digit to add to make the result an even multiple of 10. For example, my 5 digit ZIP code is 01267; these add up to 16, so a check digit of 4 is added to make the result evenly divisible by 10. The barcoded version of my ZIP code is 012674. My 9 digit ZIP code is 01267-2259; these add up to 34, so the check digit is 6. The barcoded version would be 0126722596.

In order to implement POSTNET barcodes, the parameters of the printer must be examined very carefully. My printer prints dots at the rate of 84 dots per inch, and each dot is nominally 15 to 17 mils in size. A mil is 1/1000 of an inch, and 84 dpi computes to .0119" or about 12 mils. If I divide 4 dots by 84 dots per inch, I get .0476", which is quite close to the .0475" pitch specification, and well within the tolerances. This tells me that I can use 4 print columns for each bar. The only question is, can I print the bars two dots wide with two blank dots between them, or do I need to use only one dot wide with a separation of 3 blank spots. Since a dot is 16 mils wide, but printed on 12 mil centers, two dots wide would be 28 mils wide. But the maximum allowed width is only .025, so only one dot must be used. That would give a minimum width of 15 mils, which just meets the specification. Barcodes can be printed with my printer by allocating 4 columns of dots for each bar and printing the bars using one column of dots.

How many dots tall must they be? The short bar is nominally .050" high: Dividing .050 by .012 gives about 4 dots. The actual size of the bar is  $12 \times 4 + (16 - 12)$ , which is 52 mils. .052 is well within the  $.050 \pm .010$  limits. The tall bar,  $.125 \div .012$  dpi gives 10.5 dots. Since the ratio of .125 to .05 is 10 to 4, I picked 10 dots to try. The actual size of the bar would be  $12 \times 10 + (16 - 12)$ , which is .124". (Boy, these sure are coming out nice and close.)

Ok, now that I have figured out that I need 4 or 10 dots high, separated by 3 blank columns of dots, the rest will be a matter of programming the codes. But that will have to wait till next time.

**Bit Bucket**

Dateline: CNN Science and Technology Week, 1230 EST on November 24, 1991. CNN's David French reports: "High definition television, or HDTV, promises sharp wide screen pictures for your home. Researcher in Virginia are testing six competing systems. The one they choose will offer U.S. viewers a new TV standard-of-living in the 90's." According to the report, the U.S. will chose only one standard out of six standards that are being tested. The heart of the testing program is a format converter that translates TV signals into bits. The format converter has been called a bit bucket. Those putting bits in the bucket are NHK Japan Broadcasting corp, Zenith, AT&T, Phillips, Mit, and General Instrument Corporation. The winning system is likely to be a digital system, more like computers than ever before. The report goes on to state that the first systems should be available at a price of from four to five thousand dollars in about two years.

I just discovered a new disk error. The Poly reported "Verify error! Data path ob-

structed by loose ring. Tick. Tick. Tick..." I took out the disk and looked at it. -- Sure enough, the reinforcement ring on the drive had become detached; it slid down the media and was going around and round with the disk -- smacking the heads on each revolution! Once I removed the ring the disk worked fine.

**In This Issue**

Editorial . . . . .	1
<i>PolyLetter</i> is also for	
users of System-88 . . . . .	1
Letters . . . . .	1
Bob Bybee . . . . .	1
Mitchell S. Lippman . . . . .	2
PRINT statements	
in Poly and PC BASIC . . . . .	2
Advertising . . . . .	7
HELP BASIC FORMAT . . . . .	7
HELP BASIC FUNCTIONS . . . . .	8
HELP BASIC FUNCTION STR\$ . . . . .	8
Annual Index - 1991 . . . . .	8
Postal Barcodes - part 1 . . . . .	9
Bit Bucket . . . . .	10

**PolyLetter**  
 191 White Oaks Road  
 Williamstown, MA 01267  
 (413) 458-3597

Address Correction Requested

**FIRST CLASS MAIL**



Ralph E. Kenyon, Jr.                   EXP:99#9  
 Abstract Systems, etc.               184  
 191 White Oaks Road  
 Williamstown, MA                   01267-2256

© Copyright 1991 by Ralph E. Kenyon, Jr.

PolyLetter Editor and Publisher: Ralph Kenyon. Subscriptions: US \$18.00 yr., Canada \$20.00 yr., Overseas \$25.00 yr., payable in US dollars to Ralph Kenyon. Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions, for articles, or questions you'd like answered are readily accepted. This is your newsletter; please help support it. Non-commercial subscriber adds are free of charge. PolyLetter is not affiliated with PolyMorphic Systems.

Back volumes of *PolyLetter* (1980 through 1990) are available at reduced prices payable in US dollars to Ralph Kenyon. 1 Vol. - \$15, 2 - \$28, 3 - \$40, 4 - \$50, 5 - \$59, 6 - \$67, 7 - \$75; Canada add \$3 shipping, Overseas add \$10. Individual back issues are also available (US: \$3.50, Canada: \$4.00, Overseas: \$5.00).