"We must all hang together, or assuredly we shall all hang separately."

B. Franklin concisely states why the Gazette exists: The national magazines write nothing on SuperPET. With the exception of one article in COMPUTE! and a brief and friendly review of microBASIC in the October MICRO by the competent columnist Loren Wright, SuperPET has received zero press coverage since introduction.

Any computer is defined and explored to its limits by its users, especially when the manuals are vague. Users exchange information--and learn their machines--from the computer magazines.

I frankly think the national computer magazines have ignored SuperPET because it is too rare, too complex, and too expensive. Five languages reside in it, alongside a powerful MicroEDITOR. Add SPET's ability to operate with a Host computer, and you have a most complex system.

How are isolated SPET users to help each other if the national computer press continues to concentrate where the revenue and circulation lie, with the cheap, common, machines?

So far as I know, only one other SuperPET User's Group exists--the I.P. Sharp Spet Group, which concentrates on APL. On what other resources can we draw? On infoWAT, the newsletter of Waterloo Computing Systems, Ltd., and on WATNEWS, published by the Computing Systems Group at Waterloo (and from which, praise be, we have secured much useful data). Are these sources, by themselves, adequate?

I think the answer is no. The University publishes, from its own viewpoint , general system information; infoWAT is oriented toward its own business, and prints general information on products and availability.

Which leaves us with the Commodore Customer Support Staff and Commodore Magazine, which obviously must cover every product in the Commodore line (which is growing like Topsy). So far, Walt Kutz, the U.S. SuperPET Product Manager, is confined to one page every other month. And what we submit gets screwed up.

SuperPET demands a newsletter in which users exchange information. Without the Gazette, each of you will be back groping, alone and isolated, for solutions someone else, half a nation away, has already discovered.

The best material will come from users who are learning; not from the experts (the manuals were written by experts; there's the rub. Only experts can follows them). The Gazette is meant to be a medium of interchange of information, between people learning a complex system, not a monologue. The Editor wants to edit, not to write.

In few words: (1): Contribute! Contribute! Contribute! (2) Volunteer as an Associate Editor if you are learning and can write what you learn, and can evaluate material submitted, (3) Join SPET. The ten dollars (U.S.) covers printing and postage, no more. And, (4) Comment! Suggest! Improve! Revise! CONTRIBUTE!

GOOD NEWS

We have two new Associate Editors. The first is Bob Davis of Pennington, N.J. (PASCAL). I admit to some arm twisting, since Bob is learning (good: the experts forget the hard things after they're learned). The second is Gary Ratliff of Magee, Miss., an old PET hand and experienced writer for the computer press, who will handle 6809/6502 Assembler. We hope to have articles from both next issue. Commodore has promised some public domain communications data on disk; when it is available, we will announce how to get it.

THE FEEDBACK LOOP

Summary of letters in: (1) the material is too basic, (2) it is too complex (3) not enough on languages other than microBASIC (true: get those contributions in!) and (4) stick to the objective: define the machine, so that any operator, in any language, can intelligently control his disk drives, screen, keyboard, and printer. This last came through loud and clear. And that we will do.

ADVICE NEEDED! ADVICE NEEDED!

Tony Klinkert, PO Box 996, Carrollton, Texas 75006 has printing problems with microAPL. Once his program is running ok on the screen, he has to insert a []PUT command (slow, complicated, and difficult) prior to each output to printer (8300P). He has a whole library of APL functions which previously did not need this, and must add to each the □PUT command. He asks if there's a a routine by which he can choose to output to the printer what appears on the screen, without interminable changes to that existing APL library. Write Tony if you have an answer. And send us a copy.

SUPPLEMENTARY INFO ON USING PRINTERS WITH SUPERPET

This supplements our note in issue No. 1.: Commodore software and printers use a character set called PET ASC II. Waterloo Software in SuperPET usually outputs standard ASC II to a printer. SuperPET also has two ports or busses available--IEEE and RS232. You have to employ the right printer filename for the printer and port or bus you're using. Here are the filenames to talk to any printer, from any port or bus, in any SPET language:

SuperPET to any Commodore Printer but Commodore 8300 (Diablo 630), IEEE bus: filenname "printer" (SuperPET translates to PET ASC II for the printer.)

SuperPET to any ASC II printer (including Diablo 630 and the 8300), IEEE bus: filename "ieee4" (assuming the printer is device number 4).

SuperPET to printers attached to the RS232 port: "serial". ASC II is not translated to PET ASC II when you use the filename "serial", unless you interpose your own software.

To illustrate, we apply the filenames to the print command (put) used from the microEDITOR:

Commodore	ASC II	ASC II
IEEE bus	IEEE bus	RS232 port
put "printer"	put "ieee4"	put "serial"

The three different filenames for the printer apply in all SPET languages. If MEMBERS ONLY need a sample printer program using the filenames, send the Editor a self-addressed, postpaid envelope, and tell us the language involved. Thirty-seven cents postage for all. (To give an idea of Tony Klinkert's problem, there's an APL print statement below. It prints 'message').

```
microAPL          ASC II          'IEEE4' ⎕ CREATE 1
                  IEEE            (⎕XR  'message', ⎕TC[ 7 ]) ⎕ PUT 1
                                  ⎕ UNTIE 1
```

As users become more familiar with SPET, we suspect they will use the Med more and more for printing. MeD doesn't care whether there are line numbers, or whether they are in ordinal sequence. What you see is what you print. You can limit what you print from MeD by interposing the RELATIVE LINE NUMBERS of the lines you want before the put command. To print relative lines 1 through 50, the command (ASC II printer, IEEE bus) is: 1,50 put "ieee4". Which the printer does, and stops.

In microBASIC, you list a program from the general editor with (ASCII, IEEE) save "ieee4", (not, as Oct/Nov Commodore says, by issuing the command from the MeD). Once the printer starts, nothing will stop it but a sledgehammer, pausing the printer, or cutting power. If you print long programs, or use cut paper, it's far easier to save programs from MeD with the relative line numbers and the put command.

Incidentally, for those who haven't tried them, microPASCAL and microFORTRAN are written within the microEDITOR, and a "put" is the way to list a program (or any part of it) to the printer, in those languages.

A MICROEDITOR OF LIMITED CONVENIENCE?

One of the most respected writers on PET is Loren Wright of MICRO. In the October issue, he described the MeD as much less convenient than the general editor. Only time we ever disagreed with the man. Example: Let's suppose we have a program with no room for new line numbers; we're in MeD. We want to insert 10 new lines. We press SHIFT keypad 0 ten times for ten blank spaces, and write:

```
100 proc pokegraphics
100 data 4, 40, 5, 5, 7, 33, 9, 7, 0
100 loop
100   read i
100   if i = 0 then quit
100   poke -6016, i
100   read i
100   poke -6015, i
100 endloop
100 endproc ! for proc unpoke, substitute the following data line:
              data 4, 32, 5, 3, 7, 29, 9, 9, 0
```

MeD doesn't recognize line numbers (how could it, working in FORTRAN and PASCAL?). What do you do with ten lines, all numbered 100? You say bye to the MeD, and command: renumber 100 <RETURN>, and out, neatly renumbered from 100 to 200, comes the program above in line number increments of ten. Try it. Stick the program above in the MeD and do it. Then assign RANDOM numbers, totally out of order, to the lines above, and try it again. You still get a poke proc for graphics, properly numbered.

In short, you can ignore line numbers in the MeD so long as the statements fall in logical sequence, and so long as you renumber before a run. You DO NOT have to exit the MeD and renumber a program when there are no line numbers left open.

Let's take one more step. Suppose you assemble in the MeD a new program from the bits and pieces of others. You retrieve and store on disk the old stuff by using the put command. You pick off RELATIVE LINES 200,380 from old program "whatsit" with 200,380 put "disk/1.transfer" <RETURN>. You save the rest of the old material the same way under a noted filename. Then you assemble the pieces by putting the screen cursor where you want the transfer made (data will insert on the next line below), and command: get "disk/1.transfer". MeD inserts the old material right where you want it.

Line numbers? Who cares (unless you use conditional gotos in error traps) at which point you should renumber. Anybody using gosubs in microBASIC is better off with an abacus. Trade now.

If you assemble a new program this way, from old, known, tested parts, the line numbers will resemble the random walk of an idiot. Be of good cheer. Say bye, renumber 100, and you've got a new program.

If you think us crazy for using negative numbers in a poke statement, we can only say SPET has a built-in conversion from hex to decimal; that's the way tne numbers come out; they work; why worry?

May I now ask for an opinion on the convenience of the microEDITOR? The above is only the beginning. I never use the general editor, except to renumber. It's too inconvenient; it won't scroll the program, which MeD does, not only by cursor, but by screen page, or by command to start or end of program, to a specific line number, or to a specific phrase, name, procedure, or function.

OLD PET HANDS THREATEN TAR AND FEATHERS!

Some old PET hands, with Vol. 1 No. 2 in hand, brandished cudgels and threatened to tar and feather the editor for misuse of the RETURN key:

The RETURN key in BASIC, if pressed alone during an input statement, takes you out of program. A program printed in the last issue used the RETURN key as the required input. In SPET, an input statement calling for a RETURN is a handy way to pause the program for any reason. Typically

```
input "When through reading, press RETURN: ", o$
```

will return a null string, and pause the program until you press RETURN. You throw o$ away. The method gives you substantial chances to branch within program. Example:

```
input "Press d to dump; v to line dump; RETURN alone to go on: ", o$
```

Using print statements, you can set up a whole menu on the screen this way, choose any option, or bypass 'em all. Give those feathers back to the chickens, boys (winter's comin' on).

BOBBLES, BOOBOOS & NOTES ON THIS AND THAT

The editor blew it in copies 1 through 24 of Vol. 1, No. 2 (page 3), when talking about chr$(49) being a lower case "l". The example was based on the number one; please read it so. All copies after 24 were corrected.

Scratching and Overfiling

Some of us have overfiled (updated) an existing program in the upper 64 (6809 mode) to disk, only to find--sometimes--the old program, unchanged, still on that disk. After trials, filing on two disks, we confirmed the bobble. You cannot depend on an overfiling to update an old program. You are safe only if you scratch the old program and then save the modified program.

A Comparison of Time to Execute

Two identical programs were executed: one in BASIC 4.0, the second in microBASIC. The programs ran a for-next loop from i = 1 to 500, calling a gosub or a procedure with a j = 1 to 10 loop in it. In BASIC, the subroutines were located at various distances down the program; in microBASIC a lot of unused program was stuffed in, and the working procedure was located top, bottom, and middle. Results:

Subroutine at:	BASIC 4.0	Procedure at:	microBASIC
line 1000	13.65 sec	Beginning	21 sec
line 2000	16.63 sec	Middle	21 sec
line 3000	19.33 sec	End	21 sec
line 4000	22.42 sec		

BASIC 4.0 is faster in executing a for-next loop, but when it comes to long programs, with a lot of subroutine or procedure calls, it's a changing story. The first program was changed to make four calls to the subroutine or procedure within the i = 1 to 500 loop. Results:

Subroutines at:	BASIC 4.0	Procedure at:	microBASIC
1000, 2000, 3000, 4000	70 sec	End	80 sec
One at 4000	87.63 sec	Beginning	80 sec

The results are printed for what they're worth. If any benchmark specialist wants to run a statistically valid study, we'd be happy to print the results.

THAT UGLY DUCKLING OF A DOS TURNS HALFWAY INTO A SWAN

For reasons unknown, the designers of SuperPET employed the old commands from <3.0--often easy to mess up. They are in summary form below (and we have some surprises). Rule: The command, or abbreviation, must be in CAPITALS.

(To keep examples simple: (1) drive 1 ALWAYS is source, (2) all commands are to drive 1 when one drive is used, (3) we use one id number, 22. Follow the format **rigorously!** Note CAPITALS. All editorial comments are in brackets [ ]. You can use an apostrophe ' instead of quotes.)

Now for the good news. In immediate mode, All DOS commands work in microBASIC, microFORTRAN, in the Assembler MeD, and in the microEDITOR alone. We don't have microCOBOL yet (and so don't know). In APL and PASCAL the DOS commands are so thoroughly integrated with the languages that we shall need a special article on the matter. Writers, start writing!

In microBASIC only do you need the quotation marks, open#, print#, and close# statements we show on the "NEW" command, below. In the rest of the languages follow the method outlined below, using the MeD.

Function	**Command**	
Format or "NEW" a disk	open#2, "disk", output print#2, "N1:diskname,22" close#2	[microBASIC sample]
Format or "NEW" a disk in the microEDITOR used alone, in Assembler MeD, or in FORTRAN. Do not try in MeD in microBASIC)	<beginning of program> N1:diskname,22 <end of program> put disk	[line in editor] [use no quotes!] [line in editor] [command given at command cursor]

[From here on, we omit the quotes, open, print, and close#'s to save space].

Initialize a Disk	I1	
Initialize Both Disks	I	
Validate a Disk (Same as 4.0 Collect)	V1	
Copy or Rename a File	C0:newname=1:oldname	
Copy a Whole Disk	C0=1	[drive 0 is destination]
Backup a Disk	D0=1	[drive 0 is destination]

Rename a File	R1:newname=oldname	
Scratch a File	S1:filename	
Scratch a Disk	S1:*	[leave no space between colon and *]

1. Comments and Questions on DOS Commands: The "NEW" command in the microEDITOR in the listing above is an example of how you can use ALL of the commands above in the MeD alone, MeD in Assembler and in FORTRAN, without quotes, with the "put disk" command. The DOS commands work. Here's how a "scratch a disk" command looks:

```
<beginning of file>
S1:*                   [be sure to hit RETURN]
<end of file>

put disk              [command at command cursor; hit RETURN]
```

2. Better news: suppose you have a long microFORTRAN program on screen; you want to use a DOS command. Type the command on RELATIVE line 1 (just under <beginning of file>), hit RETURN, go to the command cursor and say: 1 put disk RETURN. MeD puts relative line 1 (your command) to disk, and DOS obeys.

3. Even though microPASCAL is also written in the MeD, you cannot issue DOS commands as outlined above. All you get is syntax errors.

4. In BASIC 4.0 (and, Waterloo claims, in the various microlanguages) you cannot use the copy command to copy a whole disk to another disk UNLESS the disk numbers are identical. Fiddlesticks. Put a short (about 3-block) "dummy" file on the newed, empty disk. Then copy.

5. Copy command will stop and give an error message if a file with the same name exists on the disk copied to. Check before you try to copy.

6. When you copy or change names the old DOS allows serious errors. After wiping out a few disks, we wrote a bulletproof program. Proc dos follows. Run in 6809 mode, it works just as well on BASIC <4.0 disks and files. Just capitalize the filenames. Don't try to use it as a "high number" immediate mode procedure. The variable names may wreck your main program.

7. As to Error Handling: If an IO error stops a program, you must remember to close channels. In proc dos, microBASIC provides a handy way to tell the operator what the error is, continue program, and close it down properly. Note line 330 and those following "on ioerr". Stick a write-protect tab on a disk, and try to copy to it from another disk, using the program. You'll see the "write protect tab" error message, but the program will finish, tell you it's done, but that the copy job failed. SuperPET is superb in handling errors.

A Quickie: Ever have the red error LED on the disk drive go on, and find no way to get the error message or a green light? Try proc ds (it does what ?ds & ?ds$ does in 4.0:

```
60220 proc ds
60230 open #9, "disk", input
60240 linput #9, aa$
60250 print aa$
60260 close #9
60270 endproc
```

```
100 proc dos ! a disk utility in 6809. Works on BASIC <4.0 disks also.
105 print chr$(12);"                    MENU FOR PROC DOS" : print
110 menu$(0) = "New a Disk                 = 0"
115 menu$(1) = "Initialize a Disk          = 1"
120 menu$(2) = "Validate a Disk            = 2"
125 menu$(3) = "Copy a File                = 3"
130 menu$(4) = "Backup a Disk              = 4"
135 menu$(5) = "Rename a File              = 5"
140 menu$(6) = "Scratch a File             = 6"
145 menu$(7) = "Scratch all Files          = 7"
150 menu$(8) = "Copy All Files             = 8"
155 menu$(9) = "Change Name, Copy File = 9"
160 a = 161
165 for i = 0 to 9
170   if cursor(a) then print menu$(i)
175   a = a + 40
180 next i
185 if cursor (981) then print "CAPITALIZE BASIC <4.0 FILENAMES"
190 if cursor (641) then input "Type your option no., press RETURN. ", ms
195 if ms <0 or ms> 9 then print "Input error";chr$(11);chr$(6): goto 190
200 print chr$(12); "You chose to ";menu$(ms)
205 print : input "Are you sure? y/n: ", e$
210 if cursor (161) then print chr$(6)
215 if e$ <>"y" then print chr$(12) : a = 161 : goto 165
220 if ms=0 or ms=1 or ms=2 or ms=5 or ms=6 or ms=7
225   input "Type number of disk drive to be used: ", dr
230   if dr <0 or dr >1 then print chr$(11);chr$(6); : goto 225
235   print : input "Are you sure? y/n: ", e$
240   if e$<>"y" then print chr$(11);chr$(6);rpt$(chr$(11),3) : goto 225
245 endif
250 if ms=3 or ms=4 or ms=8 or ms=9
255   input "Type number of disk drive which is the SOURCE: ", dr
260   if dr<0 or dr>1 then print chr$(11);chr$(6); : goto 255
265   print: print "You choose drive";dr;"as SOURCE.   Double Check!"
270   input "To change drive no., type e RETURN. If drive OK, RETURN  ", e$
275   if cursor(321) then print chr$(6);chr$(10);chr$(6);chr$(10);chr$(6)
280   if e$ = "e" and cursor (241) then 255
285 endif
290 if dr = 0 then drto = 1
295 if dr = 1 then drto = 0
300 open #14, "disk", output
305 gg$ = " *":dr$= value$(dr):drto$=value$(drto)
310 b$=chr$(10)+chr$(160)+chr$(197)+chr$(216)+chr$(197)+chr$(195)+chr$(213)
315 c$=chr$(212)+chr$(201)+chr$(206)+chr$(199)+chr$(160)
320 ! above is brute force "EXECUTING" to save program space
325 bb$= b$+c$
330 on ioerr
335   print io_status$;"                          INPUT-OUTPUT ERROR!"
340   print rpt$(gg$,40)
345   print "Check for write-protect tabs or identical file names!"
350   flag = 1
355   resume next
360 endon
365 if ms = 0
370   print: input "Type new disk name: ", dname$
375   print: input "Type new id numbers or letters, NO i in front: ", id$
380   print bb$
```

```
385   print#14, "N";dr$;':';dname$;',';id$
390 elseif ms=1
395   print: input "To initialize both drives, type 2; one, type 0: ", both
400   print bb$
405   if both then print#14, "I"
410   if not both then print#14, "I";dr$
415 elseif ms = 2
420   print bb$
425   print#14, "V";dr$
430 elseif ms = 3 or ms = 9
435   print: input "Type filename to be copied; no quotes: ", file$
440   if ms = 9 then input "Type new name of file, no quotes: ", newfile$
445   if newfile$ = "" then newfile$ = file$
450   print bb$
455   print#14, "C";drto$;":";newfile$;"=";dr$;":";file$
460 elseif ms=4
465   print bb$
470   print #14, "D";drto$;"=";dr$
475 elseif ms=5
480   print : input "Type old filename, to be changed: ", file$
485   print: input "Type new filename: ", newfile$
490   print bb$
495   print#14, "R";dr$;":";newfile$;"=";file$
500 elseif ms=6
505   print : input "Type name of file to be scratched: ", file$
510   print bb$
515   print#14, "S";dr$;":";file$
520 elseif ms=7
525   print bb$
530   print#14, "S";dr$;":*"
535 else
540   print:print "You may get a copy error if id numbers on disks do not "
545   print "match. If so, put a dummy file on the destination disk."
550   print bb$
555   print#14, "C";drto$;"=";dr$
560 endif
565 for i = 0 to 9
570   menu$(i) = ""
575 next i
580 ms=-1:e$="":dr=-1:drto=-1:dname$="":file$="":id$="":newfile$=""
585 dr$="":drto$=""
590 close#14
595 print : print: print "                              PROC DOS FINISHED"
600 if flag then print "                    Operation Not Accomplished--IO Error."
605 flag = 0
610 endproc
```

****

(Note to our Canadian friends: Canada is NOT overseas. Dues: $10.00 U.S.)

APPLICATION FOR MEMBERSHIP, SUPERPET USER'S GROUP

Name:______________________________ Disk Drive: _________ Printer:_______

Address:____
Street, PO Box   City or Town   State/Province/Country   Postal ID#

Enclose Annual Dues of $10:00 (U.S.) by check or money order, made out to Secretary, SPUG. Overseas dues: $20.00 U.S. Mail to: Paul V. Skipski, Secretary, SuperPET User's Group, 4782 Boston Post Road, Pelham, N.Y. 10803, USA.

****

Newsletter published by the SuperPET User's Group (SPUG): editorial offices at PO Box 411, Hatteras, N.C. 27943. Secretary, Paul V. Skipski, 4782 Boston Post Road, Pelham, N.Y. 10803. Membership applications and inquiries to Mr. Skipski. All newsletter material to Hatteras, attn: Dick Barnes, Editor. SuperPET is a trademark of Commodore Business Machines, Inc. Contents of this newsletter copyrighted by SPUG, 1982; reprinting by permission only. SPUG members are authorized to use the material. Enclose a self-addressed, postpaid envelope with all material submitted and all inquiries requiring reply. Membership: $10.00 per year, U.S. in North America, $20.00 overseas and elsewhere. See enclosed application.

For our Canadian friends and those overseas: all nations members of the Postal Union offer certificates good in the postage of any other country for a small charge. The Union includes Canada, U.S., England, France, Germany, Russia, China, most Arabic Nations, and most European Nations. If your postmaster doesn't have one, a shillelagh is reported effective.

The SuperPET Gazette
PO Box 411
Hatteras, N.C. 27943

HATTERAS, NC
NOV 12
PM
1982
27943

USA
20c