

# Midnite Software Gazette

The First Independent U.S. Magazine for users of Commodore brand computers.

Includes The PAPER



IN THE ISSUE :

MIDNITE MEANDERINGS.....	PAGE 1
BOOK REVIEWS.....	PAGE 3
GRAPHICS REVIEWS.....	PAGE 4
HOME AND SCHOOL.....	PAGE 9
GAMES REVIEWS.....	PAGE 12
TERMINAL REVIEWS.....	PAGE 16
THE PAPER ARTICLES BEGIN AGAIN.....	PAGE 22
ARTICLE INDEX.....	PAGE 23
BRITISH MARKET.....	PAGE 26
PAPER MATE BY RILEY UPDATE.....	PAGE 27
8023 P PRINTER.....	PAGE 28
BASIC LINE CROSS REFERENCE.....	PAGE 29

I N T R O D U C I N G

# SUPER KIT/1541

BY MARTY FRANZ & JOE PETER

### SINGLE/DUAL NORMAL COPIER

Copies a disk with no errors in 32.68 seconds. dual version has graphics & music.

### SINGLE/DUAL NIBBLE COPIER

Nibble Copies a disk in 34.92 seconds. dual version has graphics & music.

### SINGLE/DUAL FILE COPIER

6 times normal DOS speed. Includes multi-copy, multi-scratch, view/edit BAM, & NEW SUPER DOS MODE.

### TRACK & SECTOR EDITOR

Full editing of t&s in hex, dec, ascii, bin. Includes monitor/disassembler with printout commands.

### GCR EDITOR

Yes disk fans, a full blown sector by sector or track by track GCR Editor. Includes Bit Density Scan.

### SUPER DOS I

Fast boot for SUPER DOS. 150 blks in 10.12 seconds.

### SUPER DOS II

Screen on and still loads 150 in 14.87 seconds.

### SUPER NIBBLER

Quite frankly, if it can be copied on a 1541 this will do it! Including Abacus, Timeworks, Accolayde, Epyx, Acti-vision, Electronic Arts.

## \$29.95

PLUS \$3.00 SHIPPING/HANDLING CHARGE -- \$5.00 C.O.D. CHARGE

**PRISM**  
SOFTWARE

401 LAKE AIR DR., SUITE D • WACO, TEXAS 76710  
ORDERS (817) 757-4031 • TECH (817) 751-0200  
MASTERCARD & VISA ACCEPTED

### GRAPHIC SCREEN CONVERTER.



By Ben Reynolds  
R & M Graphics Plus

## THE KOALA, PRINTSHOP, DOODLE, SUPER SKETCH, MICRO ILLUSTRATOR, SCREEN CONVERTER.



By Ben Reynolds  
R & M Graphics Plus

(C) 1986, Fran Co., 800 SW Jefferson, Peoria, IL 61605

(C)copyright 1986 Micro-PACE,  
All Rights Reserved

Published by: MicroPACE Computers,  
Robert Wolter.  
Editor-In-Chief: Jim Oldfield Jr.  
Editor: Tim Sickbert  
Art Director: Art Lewis Kimball  
Assoc. Editors: Robert Baker  
Dr. R. Immers  
Mike Stout  
J and E Strasma

Address for all correspondence:  
PO Box 1747  
Champaign, IL. 61820

Telephone: 217-356-1885  
BBS 217-356-8056  
(Punter 300/1200)

Issue #32 April 1986  
All contents Copyright 1986 Micro-  
PACE Inc.

Graphics used:

NEWSROOM, (TM) Springboard  
Commodore, PET, VIC-20, Commodore-64, Amiga  
Commodore 128 PC, PC-10, PC-20, B-128, are  
all copyrights or trademarks of Commodore  
Electronics, LTD. Commodore Business Machines.  
Circulation this issue: 2000

## ★ COMMODORE USERS ★

Join the largest, active Commodore users group.

### Benefit from:

— Access to hundreds of public domain  
programs on tape and disk for your  
**Commodore 64, VIC 20 and PET/CBM.**

— Informative monthly club magazine  
Send \$1.00 for Information Package.  
(Free with membership).

TPUG yearly memberships:

Regular member (attends meetings)	- \$30.00 Cdn.
Student member (full-time, attends meetings)	- \$20.00 Cdn.
Associate (Canada)	- \$20.00 Cdn.
Associate (U.S.A.)	- \$20.00 U.S.
	- \$25.00 Cdn.
Associate (Overseas - sea mail)	- \$30.00 U.S.
Associate (Overseas - air mail)	- \$40.00 U.S.

**TPUG Inc.**  
DEPARTMENT "N"

1912A Avenue Road, Suite 1  
Toronto, Ontario, Canada M5M 4A1

\* LET US KNOW WHICH MACHINE YOU USE \*





8023P PRINTER  
DEVICE NUMBER SELECTION

by Robert W. Baker

The Commodore 8023P printer has recently been showing up at extremely attractive prices. This is a 132 column, 180 character per second, IEEE-488/GPIB interface printer intended for the older PET and CBM systems. It can, however, be used with the C-64 and C-128 with an appropriate IEEE-488 interface adaptor.

Many users are acquiring 8023P's as secondary printers for their existing systems and have been asking about permanently changing the device number, as is common practice on the 1541 disk. Fear not, for it can be done in the printer and is in fact a very simple change, very similar to that on the 1541.

To open the printer, disconnect the line cord and lift the top cover as you normally would to change the ribbon. Now remove the screw at each side of the printer, at the rear of the opening just behind the paper feed mechanism. Once these two screws are removed, the entire cover can be gently lifted off the printer and placed aside. The internal logic board should now be visible in the rear compartment of the printer, with the components toward the front of the printer.

Looking at the logic board from the front, you'll notice a number of large 40 pin IC's, with the left most IC marked "U10" on the board. To the left of this IC you should see a small box marked on the board with the numbers one, two and three to the left. Inside the box are the three corresponding device number selection pads, similar to those in the 1541 disk.

Each pad consists of two metallic semicircles with a connecting trace. To change the printer's device number you simply cut the connecting wire in the appropriate pad(s). Cutting the number one

pad adds one to the normal printer device number, cutting the number two pad adds two, and cutting the number three pad adds four. The pads can be cut in any combination to select a device number between four and eleven.

To cut the trace in the pad, you should use a sharp knife like an Exacto. To re-connect a pad, if you later want to restore the printer's normal device number, simply solder a wire across the two semicircles. If you want to add an external device select switch, simply cut the desired pad and wire a single pole single throw switch across the two semicircles. The switch can then be mounted wherever convenient, but try to keep the wire length as short as possible.

Be careful when working inside the printer, use common sense. If you don't feel qualified to make the change, get a friend who is qualified to do it for you, or take the printer to your local dealer or repair shop.

When done making the desired modifications, replace the printer cover and install the two screws that keep it in place. Try a simple test like:

```
OPEN 4,5  
PRINT#4,"TEST STRING....."  
CLOSE 4
```

Where the device number in the open command (5) matches the new device number you think the printer should respond to. Obviously you'd replace the 5 with whatever device number you changed the printer to. If the printer doesn't seem to respond, then open the printer again and recheck your handy work.



BASIC PROGRAM  
LINE NUMBER CROSSREFERENCE

This handy utility program produces a cross reference list of every program line reference found in a BASIC program that was saved on disk. The program itself was designed to run on the C-128 in either 64 or 128 mode, as well as on the Commodore-64. It properly analyzes all programs written in BASIC 7.0 on the C-128 or BASIC 2.0 on the C-64 or VIC-20. It cannot handle programs written with BASIC 4.0 on the PET or CBM systems due to conflicts in the BASIC token assignments. This is an updated version of my earlier GOTO/GOSUB cross reference utility in the Commodore-64 Programmer's Library.

This program provides a list of every program line referenced by another line in your program, along with a list of every line that references that particular line. The lines being referenced are those line numbers shown as arguments for the following BASIC commands:

```
GOTO x
GO TO x
GOSUB x
ON...GOTO x,y,z...
ON...GOSUB x,y,z...
IF....THEN x
IF....THEN....:ELSE x
RESTORE x
RESUME x
TRAP x
RUN x
LIST x-y
COLLISION...,x
```

The output produced gives you a handy way of finding what subroutines or program lines are not being used, what areas are being used the most/least, etc. I've found it very handy in streamlining program size and/or execution times, or for just plain housekeeping.

When you run the program, it first asks for the filename of the program file stored on disk to be analyzed. The program

cannot be modified to read program files from cassette tape since they cannot be read as data files. Program files can, however, be read from disk as data files just as they appear in memory. To do this, the program file must be opened with a standard OPEN command, the newer DOPEN command in BASIC 7.0 cannot be used.

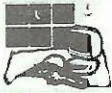
After opening the disk file, the program first reads and discards the two byte load address (end of line 200). The program file is then scanned, looking for the appropriate BASIC tokens (lines 260-690). As each token is found, the following line number(s) are read and saved in the GT matrix. The line number of the current line is saved in the LL\$ matrix to create the cross reference listing for the corresponding target line.

This program assumes the program file being read follows normal program syntax and doesn't contain any abnormal situations. In other words, the error checking is not 100% fool proof. It does, however, properly handle multiple line numbers in ON...GOTO/GOSUB constructs, periods used for line number zero, and various IF...THEN constructs.

During the scanning of the program files, the link address for each line is read in line 210 and checked for a zero value indicating the end of the program. The current BASIC line number from the start of each line is read in line 220 and converted to a decimal number in LN and a string in LN\$. Program lines are scanned by lines 260-690 and found target line numbers are added to the table by the subroutine at lines 1220-1400. Note that new entries in the matrices are inserted in their proper position to always keep the entries in ascending numerical order.

The target references are collected in the LL\$ matrix entries. The first part of every entry in this matrix is the corresponding target line number that entry refers to. Each reference line number is just added to the end of the string in each entry as it is found. If too many references occur for any particular line number, another entry is





added to the LL\$ matrix with the same line number plus 0.1 to allow more references to be added. Remember that any character string cannot exceed 255 characters in length within BASIC. So, if the list of references gets to close to this limit, another entry must be created for the new references.

As the program executes, the line number of the current line being analyzed is displayed so you can see how the program is progressing. Be patient, the program can take a while to analyze large programs or those that reference a large number of program lines. While on the subject, the program is currently limited to handling up to 500 target lines as set by the dimensions of GT and LL\$ in line 110. This seems to be a reasonable limit for most systems. You may, however, run out of memory if your program being analyzed contains an abnormally high number of line references. If you are running this program on a C-128 in 128 mode, then you can safely raise this value if needed.

Once the data is collected, you're given the option of printing or displaying the formatted information. In either case, the first line of output indicates the filename of the program that was analyzed. The left column of numbers lists the actual target lines found in the program. These are the lines referenced by the various BASIC commands within the program. Each program line containing a reference to the listed target line is then listed to the right of the separating dash. If enough reference lines were found to fill more than one line, the target line number will only appear on the first line printed.

While the output is being displayed or printed, pressing any key on the keyboard will suspend the output. This is especially convenient when using the screen display, so you can stop the screen from scrolling to read what is being generated. When ready to continue, simply press another key on the keyboard and the output will resume. If you press the 'Q'

key when the output is suspended, you can terminate the program without any further output being generated.

Screen displays are formatted for 40 column lines while printer output is formatted for 80 column lines. If you want to run this program on a C-128 with an 80 column display in 128 mode, then change the first value of RM from 25 to 65 in line 790. If your printer has more or less than 80 characters per line, then change the value of RM at the end of line 790 to 15 less than the maximum printer line length. If you need to do anything special for your printer, you can add lines before or after the OPEN in line 760.

One final note, if any errors are detected while reading the program file from disk, the error information returned from the disk will be displayed and the program will terminate with all files properly closed.

For those that don't like to type or have trouble getting the program to work, my standing offer still applies. Simply send \$5 to cover costs and I'll send a copy of the program on disk. I'll take care of the disk, mailer, and first class postage.

Robert W. Baker

Please note the following conventions in the included listing.

<u>j</u> = cursor right	<u>l/A</u> = cursor left
<u>g</u> = cursor down	<u>Q</u> = cursor up
<u>s</u> = home	<u>S</u> = clear home
<u>r</u> = reverse on	<u>R</u> = reverse off

I know the program works. It has to. I listed it right after I ran it. I know how frustrating it can be to key a program in and then not have it work correctly. And then you can proofread it again and again, and still find no error. If you cannot get it after several tries, either send \$5 to Bob Baker, or, if you have a modem, download it from Quantum Link or Starship MPC. All programs listed in the Midnite can be found in either place.

Tim



```
10 rem      basic program
20 rem      line cross references
30 rem      by robert w. baker
40 :
50 print"S";spc(13);"basic program"
60 print"      line number cross referencesqg"
70 print"prints or displays a cross reference"
80 print"table of all program line numbers used"
90 print"in any basic program saved on disk.q"
100 print"      rem 37 C= 'q'      q"
110 read nt: dim t(nt),gt(500),l1$(500)
120 for x=1 to nt: read t(x): next x
130 sp$=chr$(160): gn=0
140 print"name of basic program on disk:"
150 print"lll";sp$;"lll";
160 input fl$: if fl$=sp$ then end
170 close 15: open 15,8,15
180 open 5,8,5,"0:"+fl$+",p,r": gosub 1090
190 print"qok, scanning program file"
200 print"qat line:": gosub 1050
210 gosub 1050: if v+v1=0 then 700
220 gosub 1050: ln=v1-(256*v)
230 ln$=" "+mid$(str$(ln),2)
240 print tab(10);ln$;"      ": rem 10 spaces
250 print"q";
260 rem scan basic line for references
270 gosub 1060
280 if v=32 then 270
290 if v=0 then 210
300 if v<>34 then 350: rem skip inside quotes
310 gosub 1060: if v=34 then 270
320 if v>0 then 310
330 goto 210
340 rem scan standard tokens in tbl
350 z=0: for x=1 to nt
360 if v=t(x) then z=v: x=nt
370 next x:if z>0 then gosub 1220:goto 280
380 if v<>145 then 450: rem 'on'
390 gosub 1060: if v=0 then 210
400 if (v<>137) and (v<>141) then 390
410 gosub 1220
420 if v=44 then 410
430 if v=32 then gosub 1060: goto 420
440 goto 280
450 if v<>203 then 490: rem 'go'
460 gosub 1060: if v=32 then 460
470 if v=164 then gosub 1220
480 goto 280
490 if v<>155 then 540: rem 'list'
500 gosub 1220
510 if v=32 then gosub 1060: goto 510
520 if v=171 then gosub 1220
530 goto 280
540 if v<>206 then 570: rem 2 byte tokens
550 gosub 1060: if v=0 then 210
560 goto 270
570 if v<>254 then 270
580 gosub 1060: if v=0 then 210
590 if v<>23 then 270: rem 'collision'
600 gosub 1060: if v=0 then 210
610 if v=58 then 270
620 if v=44 then gosub 1220: goto 280
630 if v<>40 then 600
640 rem ignore inside ( )
650 gosub 1060: if v=0 then 210
660 if v=58 then 270
670 if v<>41 then 650
680 goto 600
690 rem *** output results ***
700 close 5: close 15
710 print: pd=3:
720 print"qdone, want printed output (y/n): ";

'730 get c$: if c$="n" then 760
740 if c$<>"y" then 730
750 pd=4: gosub 1160
760 open 4,pd
770 gosub 1130
780 ifgn=0thenprint#4,"no refnrcs found":goto1020
790 rm=25: if pd=4 then rm=65
800 for x=0 to gn-1: if pd=3 then 820
810 ifpg=56thenfory=1to10:print#4:nexty:gosub1140
820 s$=right$(" "+str$(gt(x)),5)+" -"
830 if s$<>l$ then l$=s$: goto 850
840 s$="      ": rem 7 spaces
850 print#4,s$;
860 b=0: for y=0 to int(len(l1$(x))/rm)
870 a=b+1: b=a+rm: if b>255 then 920
880 c$=mid$(l1$(x),b,1): if c$="" then 920
890 c=asc(c$)
900 if c>47 and c<58 then b=b+1: goto 880
910 rem break line at space
920 ln$=mid$(l1$(x),a,b-a): if ln$="" then 950
930 if y>0 then print#4,"      "":rem 8 spaces
940 print#4,ln$: pg=pg+1
950 next y
960 get c$: if c$="" then 1010
970 if pd=4 then gosub 1190
980 get c$: if c$="" then 980
990 if c$="q" then 1020
1000 if pd=4 then gosub 1160
1010 next x
1020 close 4: end
1030 rem ***** subroutines *****
1040 s$=s+c$: goto 1060
1050 gosub 1060: v1=v
1060 get#5,c$: gosub 1090
1070 if c$="" then v=0: return
1080 v=asc(c$): return
1090 input#15,en,em$,et,es: if en=0 then return
1100 print:print"rdisk error:R";en;
1110 print"  trk/sec:R";et;"7";es: print em$
1120 close 4: close 5: close 15: end
1130 if pd=3 then print"S";
1140 print#4,"line# references in: ";fl$
1150 print#4: pg=2: return
1160 print"Printing cross reference tableqg"
1170 print"press any key to suspend output"
1180 return
1190 print"qgoutput suspendedqg"
1200 print"r any key to continue, q to quitr"
1210 return
1220 lt=-1
1230 gosub 1060: if v=32 then 1230
1240 if v<48 or v>57 then 1270
1250 if lt<0 then lt=0
1260 lt=(10*lt)+val(c$): gosub 1060: goto 1240
1270 if v=46 then if lt<0 then lt=0: gosub 1060
1280 if lt<0 then return
1290 z=gn: if gn=0 then 1390
1300 for x=0 to gn-1: if int(lt)<>gt(x) then 1350
1310 if right$(l1$(x),len(ln$))=ln$ then 1340
1320 if len(l1$(x))>246 then lt=lt+0.1: goto 1350
1330 l1$(x)=l1$(x)+ln$
1340 x=gn: next x: return
1350 if lt>gt(x) then next x: goto 1390
1360 z=x: for y=gn to z step -1
1370 gt(y+1)=gt(y): l1$(y+1)=l1$(y): l1$(y)=""
1380 next y
1390 gt(z)=int(lt): l1$(z)=l1$(z)+ln$
1400 gn=gn+1: return
1410 data 8: rem #standard tokens
1420 rem standard tokens in following order,
1430 rem goto,gosub,then,restore
1440 rem resume,trap,run,else
1450 data 137,141,167,140,214,215,138,213
```