

KC 99'er CONNECTION

A KANSAS CITY PUBLICATION



* Volume 7 KC 99'er BBS 436-9074 Issue 3 *

MARCH ISSUE 1988

DISK OWNER HINTS (reprinted from Bus-Bytes 1984)

Have you ever had an outstanding program load into your /4A and find it just will not run. The computer keeps telling you 'MEMORY FULL AT LINE XXX'? The CALL FILES command was designed to solve this problem. A disk operating system reserves 534 bytes of computer memory for the controller. Each open file removes an additional 518 bytes of memory.

You have a maximum of 9 files available on your /4A. The default or normal is 3 open files. The following chart shows how much memory each open file removes:

CALL FILES NUMBER	MEMORY AVAILABLE	EXPANSION MEMORY
1	12876	24488
2	12358	24488
3	11840	24488
4	11322	24488
5	10804	24488
6	10286	24488
7	9768	24488
8	9250	24488
9	8732	24488

XBASIC has a size command within the module. This was used to determine how large memory was for each additional file opened.

FILES is a machine subprogram used to set the amount of open files as desired. The command format is shown as: CALL FILES(X) NEW

X must be any single digit from 1 to 9 specifying the number of files to be open. The NEW command forces the computer to implement the CALL FILES command entered previously. Using CALL FILES in a program is NOT RECOMMENDED. A failure to type NEW after it can cause the machine to lock up and other unpredictable results.

CALL FILES is cancelled by turning off the computer. Next time you encounter a 'MEMORY FULL IN LINE XXX' save your program and then use CALL FILES. The command works in both Basic and Xbasic.

Computer-Puzzle Clues

Across

- 1A Spread-----
- 1G CLS in Applesoft BASIC
- 1L Type style
- 2A Word with communicate or vision
- 2F Bar used to raise pitch of strined instruments
- 2K Made of flax
- 3A Banish
- 3G Unexpected
- 3N Spielbers character
- 4A Anatomical vessel
- 4E Egyptian sun-god
- 4H Disk division
- 5F Mr. Lewis of sci-fi
- 5M Recede; abate
- 6A Computer hotbed; two words with 6J Across
- 6J See 6A Across
- 7B Prefix for host or exist
- 7E What an Epson FX-80 can do
- 7K Smallest amount
- 8A Portion of land
- 8E Command to reverse last action
- 8L Feminine suffix
- 9A -----date; overwhelm
- 9G Wicked
- 9L Grande or de Janeiro
- 10C Compass direction
- 10H Superlative suffix
- 10M Possess
- 11A Symbol for technetium
- 11E Electronic drawing pad or Australian marsupial
- 12A Sneaky
- 12E Computer corp.
- 12I Clan
- 12N Opposite of out
- 13B Letters in a ligature
- 13E System of symbols
- 13J Equality
- 14A Andy's sidekick
- 14E Word with skeleton or skate
- 14I Fundamental; computer language
- 15A Trial
- 15K Rodent

Down

- 1A Wozniak and Jobs
- 1B -----decimal
- 1C Ivy League bulldog
- 1D Moray
- 1G Owns
- 1H Musical composition
- 1I A la -----
- 1L End. to Guido

- 1M Not off
- 1N Born, in Nice
- 1O Explosive
- 2K Common BASIC keyword, often omitted
- 3E Pause in speech
- 3J City belongs to no state, abbr.
- 4F Oak nut
- 4M Kin
- 5D Fa share
- 5G Slyly disappearing
- 5N Give
- 5O Eight bits
- 6B Symbol
- 6C Type of water lily
- 6E Heart of a computer
- 6K Alabama, abbr.
- 4L Lascivious look
- 7H New
- 8A Prints a program
- 8J Key below shift on IBM PC
- 9D State with Lincoln as capital
- 9I One thing ----- good as another (2 words)
- 10D The Big Apple
- 11B Copy
- 11E Gun's recoil
- 11F Woodwind instrument
- 11G Airmail delivery, abbr.
- 11K Muscle contraction
- 12C Yearns
- 12J Government agency
- 12L Threesome
- 12N Pronoun
- 13M Place for patients in critical condition
- 14A IBM PC model with "Advanced Technology"

Computer-Puzzle Solution
=====

ABCDEFGHIJKLMNO

- 1 |SHEET*HOME*FONT|
- 2 |TELE*CAPO*LINEN|
- 3 |EXILE*SUDDEN*ET|
- 4 |VAS*RA*SECTOR**|
- 5 |E**F*CS*****EBB|
- 6 |SILICON**VALLEY|
- 7 |*CO*PRINT*LEAST|
- 8 |LOT*UNDO*A*ETTE|
- 9 |INUN**EVIL*RID*|
- 10|S*SE***EST**DOWN|
- 11|TC**KDALA*S*N*Y|
- 12|SLY*IBM*SEPT*IN|
- 13|*GE*CODE*PARITY|
- 14|ANN*KEY*BASIC**|
- 15|TEST*****MOUSE|

MULTI-PLAN TIPS

by Steve Zimmerman (reprinted from Mid-South NL)

One of the most useful features of Multiplan is WINDOWS! You know windows are a big thing now. Just look at magazine reviews and ads for programs such as Framework or Symphony. Windows enable you to see two parts of the screen, in different areas of the worksheet, all at the same time. Sounds simple, but what good is it? Well, let's say you have a worksheet about 60 rows of labels in column 1, and that you enter data for every day in a month in columns 2-32. Since your basic column widths show only 4 columns, you can see the labels only while entering the first 3 days data. If you make the columns narrower, of course you can see more if the numbers are not too long. If the numbers are too long, you will just see "*****", an error symbol that means you have too many numbers to display in that column width. To enable you to enter later columns of data in the proper rows, you need to create a window.

To do this, position your cursor (the cell pointer) in R1C2 (one column to the right of your labels) and hit the W key (for windows). On the next menu, you want to key S (for split). On the next menu key V (for vertical) and Multiplan will respond with "at column 2", linked yes no with the NO highlighted. Use tab (CTRL 2) to move the command cursor from the 2 in "at columns" to the linked field, key Y to link the windows, and then key Enter. With practice this command sequence becomes W S V tab Y Enter. You now have a window, linked vertically. As you move the cell pointer down in the active window (the one just created), the labels move along. As you move the cell pointer to the right, columns will disappear on the left side of the second window, but the labels will remain in view.

To "uncreate" this window, key W (for window) C (for close); Multiplan will respond with *2, key Enter, and the window closes. You can do the same thing with horizontal windows. If you have labels across the top and you want to see them as you move up and down.

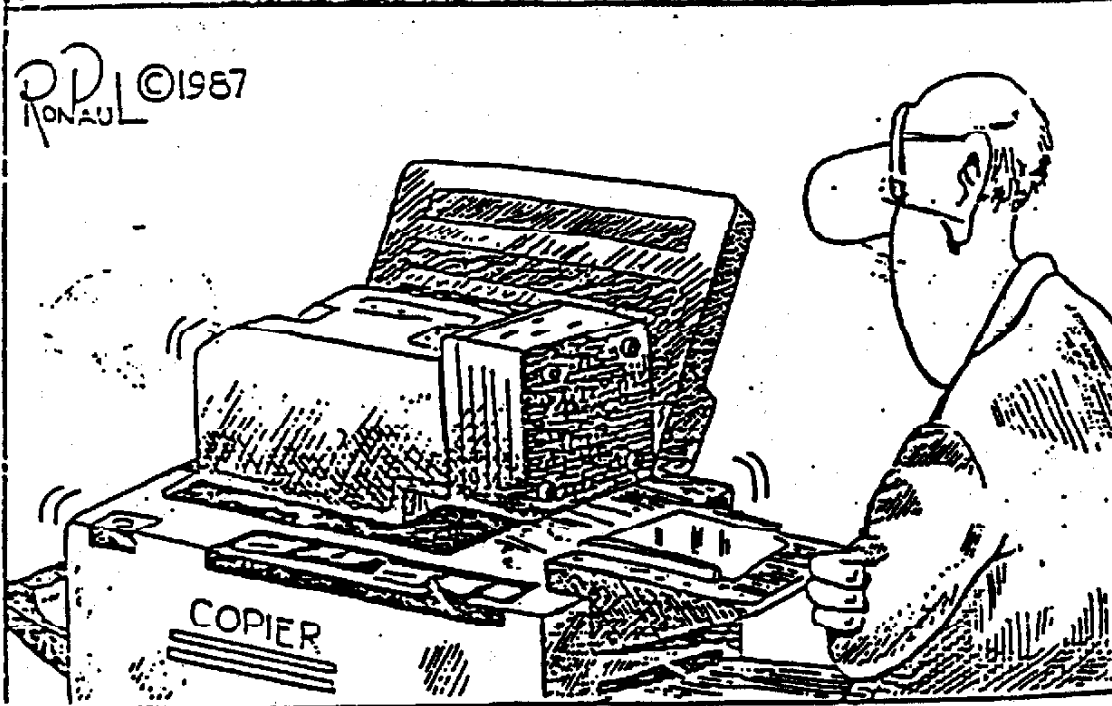
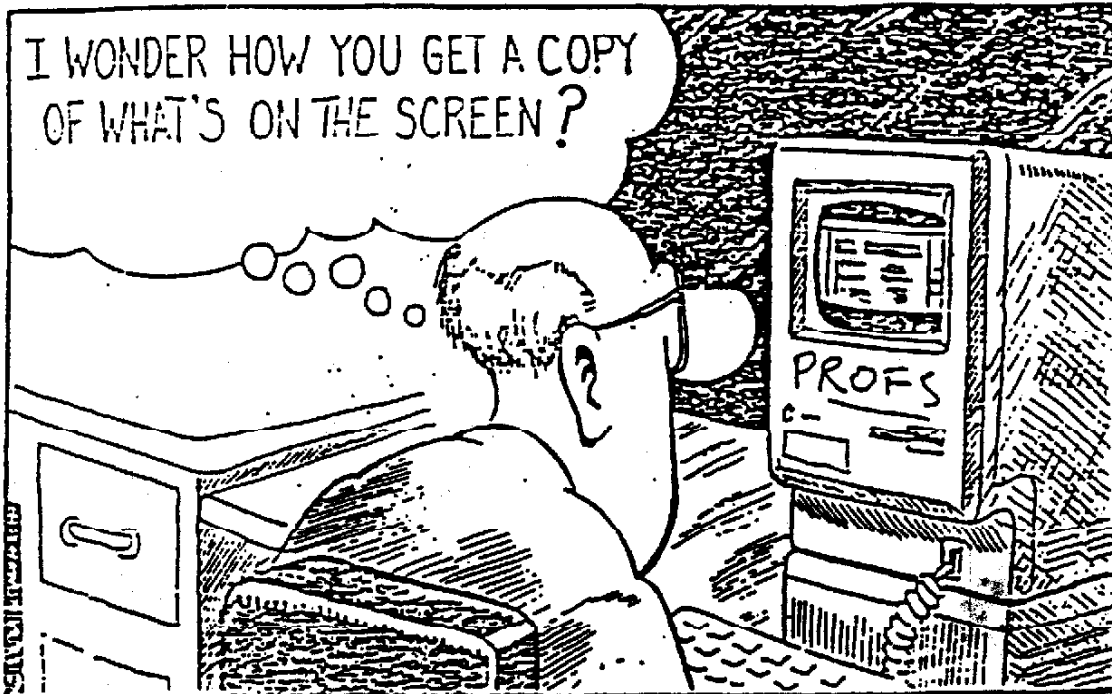
You can move the cell pointer between windows by using CTRL 6 (change window), which makes a different window the active window. CTRL 6 again will take you back to the original window.

There is a way to keep your row labels on the left in view and also keep your column labels in sight. To do this, you must use the Title Option, rather than the vertical or horizontal options. The command sequence is Window, Split, Title.

Before you enter this, make sure your cell pointer is in the column to the right of your label column and in the row below the row which contains your column labels. Multiplan will propose the number of rows and columns to be contained in the right place. Just enter the appropriate numbers of rows, tab over, enter the appropriate number of columns, and hit Enter. Multiplan will create 4 windows, numbered clockwise from the upper left, 1 2 3 4. Window 3 will be the active window, containing your data. You can now move to any row or column of your worksheet and still be able to see your row and column labels. This is a great help in entering data into a large spreadsheet.

To sum it up, Split lets you create multiple windows, Border lets you highlight or separate windows visually. Close lets you uncreate windows and Link lets you set up or break links between windows. You can split your screen into a max of 8 windows (if you can set them to fit and still show you anything). You can link windows to make them scroll together and you can use the Title command to show row and column labels to make data entry easier on a large spreadsheet. To move from the active window to another, use CTRL 6 (change window).

The ability to use multiple windows is one of the most powerful features of Multiplan and I hope that this 'raises the shades' for those of you with windowing questions.!



(reprinted from Tic Tac U.G. Newsletter)

Reprinted from NET 99ER

I/O PORT PIN ASSIGNMENTS GROM PORT

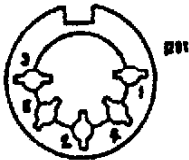


PIN	DESCRIPTION	PIN	DESCRIPTION
1	RESET	2	GND (SYSTEM)
3	D7	4	CRU CLK
5	D6	6	CRU IN
7	D5	8	A16/CRU OUT
9	D4	10	A13
11	D3	12	A12
13	D2	14	A11
15	D1	16	A10
17	D0	18	A9
19	+5 VOLT	20	A8
21	ES (GROM SELECT)	22	A7
23	MD/A14	24	A3
25	NI (DIR)	26	A6
27	GROM CLOCK	28	A5
29	-5 VOLT	30	A4
31	GR (GROM READY)	32	WE
33	GND (GROM)	34	ROM G
35	GND (SYSTEM)	36	GND (SYSTEM)

I/O PORT PIN ASSIGNMENT PERIPHERAL I/O port

PIN	DESCRIPTION	PIN	DESCRIPTION
1	+5 VOLT	2	SBE (SPEECH SELECT)
3	RESET	4	EST INT
5	A5	6	A10
7	A4	8	A11
9	TRIN	10	A2
11	A12	12	READY/HOLD
13	LOAD	14	A8
15	A13	16	A14
17	A7	18	A9
19	A15	20	A2
21	GND	22	CRU CLK
23	GND	24	D 3
25	GND	26	WE
27	GND	28	MBE
29	A6	30	A1
31	A0	32	MEMEN
33	CRU IN	34	D7
35	D4	36	D6
37	D0	38	D5
39	D2	40	D1
41	HOLD/IAQ	42	D3
43	-5 VOLT	44	SPEECH

I/O PORT PIN ASSIGNMENTS VIDEO JACK



PIN	DESCRIPTION
1	+12 VOLT SUPPLY FOR EXTERNAL UNITS SUCH AS MODULATOR
2	SHIELDING CONNECTION
3	SOUND OUTPUT
4	COMPOSITE VIDEO OUTPUT
5	GROUND CONNECTION

I/O PORT PIN ASSIGNMENTS REMOTE WIRED MACHOLD CONTROLS I/O PORT

PIN	DESCRIPTION
1	NOT CONNECTED
2	JOYSTICK B
3	KEY 0 (UP)
4	KEY 4 (PUSH BUTTON)
5	KEY 3 (LEFT)
6	NOT CONNECTED
7	JOYSTICK A
8	KEY 1 (DOWN)
9	KEY 2 (RIGHT)



I/O PIN ASSIGNMENT POWER RECEPTACLE (USHN)

PIN	DESCRIPTION
1	NOT USED
2	10 VOLT AC
3	COMMON
4	5 VOLT AC

I/O PORT PIN ASSIGNMENTS CASSETTE I/O PORT



PIN	DESCRIPTION
1	CS1 MOTOR CONTROL (POS)
2	CS1 MOTOR CONTROL (NEG)
3	GND (SYSTEM)
4	SOUND OUT
5	RECORD OUTPUT
6	CS2 MOTOR CONTROL (POS)
7	CS2 MOTOR CONTROL (NEG)
8	AUDIO IN
9	AUDIO GROUND

SOME USEFUL SUBROUTINES
from Pomona Valley Newsletter

YOUR ETSY-BITSY CATLOGGER
(from the Aminion Collection)

```
200 ! SCREEN WIPES
210 GOSUB 280
220 FOR I=24 TO 1 STEP-1 :: DISPLAY
    AT(I,1):"" :: NEXT I
230 GOSUB 280
240 CALL HCHAR*1,1,32,768)
250 GOSUB 280
260 CALL VCHAR(1,1,32,768)
270 END
280 FOR X=1 TO 24 :: PRINT "S C R E E
    N F I L L E R" :: NEXT X :: FOR
    DE=1 TO 100 :: NEXT DE :: RETURN

300 !BLINKING MESSAGE
310 CALL CLEAR
320 CALL KEY(0,K,S) :: DISPLAY AT(12,
    8):"PRESS ANY KEY" :: DISPLAY AT
    (12,8):" " :: IF S=0 THEN 320

400 !STOP SCROLLING
410 FOR X=1 TO 30 :: PRINT "PRESS ANY
    KEY" :: GOSUB 430 :: NEXT X
420 END
430 !STOP ROUTINE
440 CALL KEY(3,K,S)
450 IF S=0 THEN 490
460 CALL KEY(3,K,S)
470 IF X=0 THEN 460
480 IF S=-1 THEN 460
490 RETURN

500 !BORDER ROUTINE
510 CALL CLEAR :: CALL SCREEN(6) ::
    CALL VCHAR(1,31,1,96)
520 FOR SET=1 TO 12 :: CALL COLOR(SET
    ,13,4) :: NEXT SET
530 DISPLAY AT(12,2):"LINE 510 -BORDE
    R COLOR" :: DISPLAY AT(14,2):"LIN
    E 520 -CHARS & SCREEN"
540 GOTO 540

10000 !UNDERLINE SUBROUTINE
10010 SUB UL(R,C,Z)
10020 !(ROW,COL,SPACES)
10030 CALL CHAR(95,"00FF")
10040 FOR I=1 TO Z :: CALL HCHAR(R+1,
    C+2,95) :: C=C+1
10050 SUBEND
```

THIS IS A GOOD ONE TO KEEP AS A
LOAD ON YOUR UTILITY DISK FOR
MAKING A QUICK HARDCOPY OF YOUR
DISKETTE (editor)

```
10 !BY Warren Asee
20 @=1 :: +=2 :: OPEN #@:"PIO.LF" :
: PRINT #@:CHR$(27)&"A"&CHR$(7):CHR
$(10):CHR$(10):: CLOSE #@
30 CALL SCREEN(15):: DISPLAY AT(10,
@)ERASE ALL:"***ITSY-BITSY CAT-FRIN
T***": : : : "INSERT DISK IN DRIVE
1: THENPRESS ENTER OR 'E' TO END..
" :: ACCEPT AT(16,20):A# :: IF A#="
E" THEN CALL CLEAR :: END
40 DIM B$(127),C$(127),A(127),B(127
),D$(127),C(127),E$(5),F$(127):: E#
(@)="DIS/FIX" :: E#(+)="DIS/VAR" ::
E$(3)="INT/FIX"
50 E$(4)="INT/VAR" :: E$(5)="PROGRA
M" :: D=# :: UPEN #+:"DSK1.",INPUT
,RELATIVE,INTERNAL :: INPUT #+ :G$,E
,E,F :: FOR G=@ TO 127 :: INPUT #+ :
C$(G),A(G),B(G),D(G) :: D$(G)=STR$(B
(G)) :: IF B(G)=[] THEN D$(G)=" "
60 IF LEN(C$(G))=[] THEN 100
70 IF ABS(A(G))<>5 THEN B$(G)=" "&S
TR$(C(G))
80 IF A(G)<@ THEN F$(G)="Y"
90 NEXT G
100 CLOSE #+ :: H=INT(G/+) :: OPEN #
@:"PIO.LF" :: PRINT #@:CHR$(27)&"S"
&CHR$(1);CHR$(27)&"A"&CHR$(7);CHR$(
15);CHR$(27)&"C"&CHR$(15);CHR$(27)&
"N"&CHR$(4)
110 PRINT #@:"DSK1 - DISKNAME: ";G$
;" AVAILABLE= "; "USED=";E-F;CHR$(10
):: FOR I=@ TO H :: PRINT #@:C$(I);
TAB(12);B(I);TAB(17);E$(ABS(A(I)));
TAB(28);F$(I);
120 PRINT #@:TAB(30);C$(I+H);TAB(42
);D$(I+H);TAB(47);E$(ABS(A(I+H)));T
AB(58);F$(I+H);CHR$(10):: NEXT I ::
PRINT #@:CHR$(12):: CLOSE #@ :: GO
TO 30
```

#####

#####

IMPORTANT MESSAGE FOR ALL
COMING APRIL 10 1989
OUR FOURTH YEARLY TI SWAP-N-SHOP

LET YOUR FRIENDS KNOW

PROGRAMS THAT WRITE PROGRAMS

Part 2

by Jim Peterson

Last month I promised you something more useful, so here it is. This routine will come in very handy for formatting screen text into neat 28-column lines, and will save the text in program lines of DATA statements. When you are ready to save, type @@@ and enter as the last line, then NEW and MERGE DSK1.LINEFILE -

```

100 !LINEWRITER to aid in fo
rmatting screen text into 28
-column format and saving it
as DATA program lines in ME
RGE format - by Jim Peterson
110 !strings contains comm
as and quotation marks will
be ACCEPTed, and converted t
o DATA statements which RUN
correctly even though they
120 !are not enclosed in qu
otation marks!
130 CALL CLEAR :: OPEN #1:"D
SK1.LINEFILE",VARIABLE 163 :
: LN=30000
140 FOR R=1 TO 24 :: DISPLAY
AT(R,1)SIZE(1):" " :: ACCEP
T AT(R,0)SIZE(-28):A$ :: IF
A$="@@@" THEN 180 :: B$=B$&C
HR$(200)&CHR$(LEN(A$))&A$
150 X=X+1 :: IF X/4=INT(X/4)
THEN 160 ELSE B$=B$&CHR$(179
):: GOTO 170
160 GOSUB 210 :: LN=LN+10
170 NEXT R :: X=0 :: CALL CL
EAR :: GOTO 140
180 IF B$="" THEN 200 :: IF
SEG$(B$,LEN(B$),1)=CHR$(179)
THEN B$=SEG$(B$,1,LEN(B$)-1)
190 GOSUB 210
200 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1 :: END
210 PRINT #1:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&B$&CHR$(0):: B$=NU
L$ :: RETURN

```

Oh that puzzle in last month's article? Try creating those DATA statements with this LINEWRITER program!

Now, let's get down to business and learn how to do all

this. First, let's write a program that will write a program to list the token codes that you need to use to write a program that will write a program -

```

100 OPEN #1:"DSK1.TOKENLIST"
,DISPLAY ,VARIABLE 163,OUTPU
T :: FOR N=129 TO 254 :: L1=
INT(N/256):: L2=N-256*L1
110 PRINT #1:CHR$(L1)&CHR$(L
2)&CHR$(131)&CHR$(N)&CHR$(0)
:: NEXT N
120 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1 :: END

```

Key that in, RUN it, then enter NEW, then MERGE DSK1.TOKENLIST. Now LIST it and you will see a list of ASCII codes 129 through 254 and their token meanings. Delete lines 171 through 175, 185, 198, 226 through 231, and 242. Change the definition of 199 to QUOTED STRING, of 200 to UNQUOTED STRING, and 201 to LINE NUMBER, and add line 255 !END OF FILE.

You don't need all those exclamation points, so change the program to a DIS/VAR 80 file by LIST "DSK1.TOKENLIST". Then key in this little routine.

```

100 OPEN #1:"DSK1.TOKENLIST"
,INPUT :: OPEN #2:"PID" for
whatever
110 PRINT #2:CHR$(27);"N";CH
R$(6)
120 LINPUT #1:A$ :: PRINT #2
:TAB(10);SEG$(A$,1,4)&SEG$(A
$,6,255):: IF EOF(1)<>1 THEN
120 ELSE CLOSE #1 :: END

```

RUN it, and print out a list of all the token codes. Keep it handy, you'll be needing it. Notice that every Extended Basic statement has its own ASCII token code - even the ones you perhaps never heard of, such as LET and GO. Notice also that every keyboard symbol which affects program execution, such as + and =, has its own ASCII token code which is NOT the same as its keyboard ASCII code. And notice that the double colon, used as a separator in Extended

Basic multi-statement lines, has its own token.

Now, let's take a look at how a MERGE format program is put together. This routine will do that for you - and you will also find it very useful in debussing the MERGE programs you are going to write.

```
100 DISPLAY AT(3,5)ERASE ALL
:"D/V 163 FILE READER": :
    by Jim Peterson": : : "T
o edit a file saved or": "cre
ated in MERGE format."
110 DISPLAY AT(12,1):"Output
to? (S/P)S": " (S)screen": " (
P)rinter" :: ACCEPT AT(12,17
)SIZE(-1)VALIDATE("SP"):Q$
120 IF Q$="P" THEN DISPLAY A
T(14,1):"PRINTER? PIO" :: AC
CEPT AT(14,10)SIZE(-18):P$ :
: D=2 :: OPEN #2:P$
130 DATA ELSE,":",!,IF,GO,G
OTO,GOSUB,RETURN,DEF,DIM,END
,FOR,LET,BREAK,UNBREAK,TRACE
140 DATA UNTRACE,INPUT,DATA,
RESTORE,RANDOMIZE,NEXT,READ,
STOP,DELETE,REM,ON,PRINT,CAL
L
150 DATA OPTION,OPEN,CLOSE,S
UB,DISPLAY,IMAGE,ACCEPT,ERRO
R,WARNING,SUBEXIT,SUBEND,RUN
,LINPUT
160 DATA ,,, ,THEN,TO,STEP,"
,",";",";","),(&,,OR,AND,XOR
,NOT,=,<,>,+,-,*,/,^,
170 DATA QUOTED STRING,UNQUO
TED STRING,LINE NUMBER,EOF,A
BS,ATN,COS,EXP,INT,LOG,SGN,S
IN
180 DATA SQR,TAN,LEN,CHR$,RN
D,SEC$,POS,VAL,STR$,ASC,PI,R
EC,MAX,MIN,RPT$,,, ,,, ,NUMERI
C,DICIT
190 DATA UALPHA,SIZE,ALL,USI
NG,BEEP,ERASE,AT,BASE,,VARIA
BLE,RELATIVE,INTERNAL,SEQUEN
TIAL,OUTPUT,UPDATE,APPEND
200 DATA FIXED,PERMANENT,TAB
,#,VALIDATE
210 DIM T$(126):: FOR J=1 TO
126 :: READ T$(J):: NEXT J
:: E$(1)="LINE NOT CLOSED WI
TH CHR$(0)"
220 DISPLAY AT(16,1):"FILENA
ME? DSK" :: ACCEPT AT(16,14)
:F$
230 ON ERROR 240 :: OPEN #1:
"DSK"&F$,VARIABLE 163,INPUT
:: GOTO 250
```

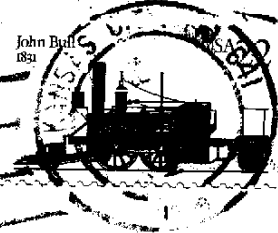
```
240 DISPLAY AT(20,1):"I/O ER
ROR" :: ON ERROR STOP :: RET
URN 220
250 ON ERROR 260 :: LINPUT #
1:A$ :: X=ASC(SEG$(A$,1,1)):
: Y=ASC(SEG$(A$,2,1)):: IF X
#255 AND Y=255 THEN 410 ELSE
270
260 PRINT #D:"FILE NOT CLOSE
D PROPERLY": "WITH CHR$(255),
CHR$(255) ?" :: STOP
270 PRINT #D:"LINE NUMBER":X
;"TIMES 256=";X*256;Y;"PLUS"
;Y;"=";X*256+Y
280 FOR J=2 TO LEN(A$)-1 ::
X=ASC(SEG$(A$,J,1))
290 IF X=201 THEN PRINT #D:X
;"LINE NUMBER" :: X=ASC(SEG$
(A$,J+1,1)): Y=ASC(SEG$(A$,
J+2,1)):: J=J+2 :: PRINT #D:
X;"TIMES 256=";X*256;Y;"PLUS"
;Y;"=";X*256+Y
300 IF X=199 THEN PRINT #D:X
;"QUOTED STRING" ELSE IF X=2
00 THEN PRINT #D:X;"UNQUOTED
STRING" ELSE GOTO 360
310 J=J+1 :: X=ASC(SEG$(A$,J
,1)):: PRINT #D:X;"OF";X;"CH
ARACTERS"
320 ON ERROR 340 :: FOR L=1
TO X :: Y=ASC(SEG$(A$,J+L,1)
):: PRINT #D:Y;CHR$(Y):: IF
Y<32 OR Y>126 THEN PRINT #D:
"UNPRINTABLE CHAR - ERROR?"
330 NEXT L :: J=J+X :: GOTO
370
340 PRINT #D:"ERROR! INSUFFI
CIENT BYTES IN": "STRING" ::
IF ASC(SEG$(A$,LEN(A$),1))<>
0 THEN PRINT #D:E$(1)
350 ON ERROR STOP :: RETURN
250
360 IF X<129 THEN PRINT #D:X
;CHR$(X): " VARIABLE NAME" FI
SE PRINT #D:X;T$(X-128)
370 CALL KEY(0,K,S):: IF S=0
THEN 390
380 CALL KEY(0,K2,S2):: IF S
2<1 THEN 380
390 NEXT J :: IF ASC(SEG$(A$
,J,1))=0 THEN PRINT #D:"0 EN
D OF LINE" ELSE PRINT #D:E$(
1)
400 GOTO 250
410 PRINT #D:X;X;"END OF FIL
E" :: CLOSE #1 :: STOP
```

Next month - how to do it!

KANSAS CITY TI99/4A COMPUTER USERS' GROUP
POST OFFICE BOX 12591
NORTH KANSAS CITY, MISSOURI 64116

FIRST CLASS MAIL

The K.C. TI99/4A Users' Group is ~~in~~ no way affiliated with or sponsored by TI. Mention of a company or product is not considered an endorsement. Opinions expressed are those of the writers and not necessarily those of the Group, its officers, editor and the membership. Information from this Newsletter may be reproduced only if credit is given to both the author and this newsletter.



| NEWSLETTER ONLY MEMBERSHIP \$12/yr |
USER GROUP MEMBERSHIP IS \$18/yr

IN THIS ISSUE

BLOODBANK PROGRAM TIPS
WRITING PROGRAMS THAT WRITE PROGRAMS
MULTIPLAN TIPS

FJ88NQSAJMB7
DALLAS TI HOME
Computer Group
P. O. Box 29863
Dallas TX 75229

```
#####  
# #  
# #  
# OUR NEXT MEETING #  
# SUNDAY - MARCH 13 1988 #  
# 2:00 P.M. #  
# Arthur Mas Center behind M.R.I. #  
# Volker Boulevard & Rockhill Road #  
# #  
# XBASIC CLASS CONTINUES #  
# #  
# #  
#####
```

-1015