

TI - D - BITS

PHILADELPHIA AREA USERS GROUP NEWSLETTER
COVERING THE TI99/4A
AND MYARC 9640 COMPUTERS

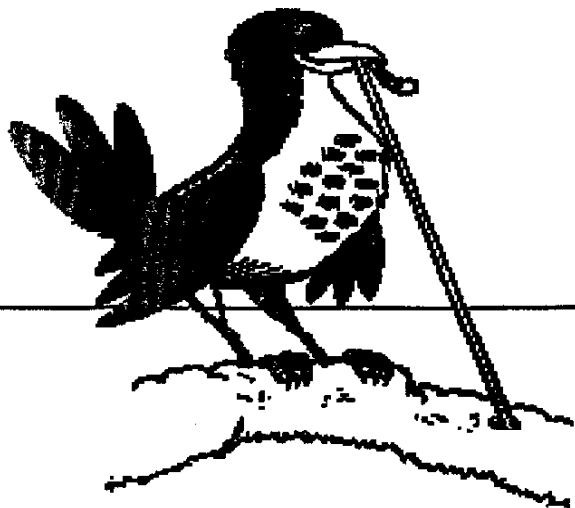
JUNE 1990

Volume 10 Number 6

THE EARLY BIRD GETS THE WORM

So Don't Let Any Grass Grow Under Your Computer

Come Out and Support your User Group and
set More out of your TI99/4A and/or GENEVE



THE PHILADELPHIA AREA TI-99/4A USERS' GROUP (JUN '90)

The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the first Saturday of any given month, we meet at the Bucks County Youth Development Center, (YDC, which is next to Meshaminy Mall), Administration Building, beginning at 10:00 am. On the third Saturday of each month, we meet at LaSalle University, 20th Olney, in room H-329 located in the Science Building. Membership to The Philadelphia Area TI-99/4A Users' Group is available to all. We invite anyone that is interested in the TI-99/4A to visit us. Stop in and see what is available to you for your TI and how membership can benefit you!

Current executive board consists of:

PRESIDENT..... Allan Silversteen. 215-885-7910
VICE PRESIDENT..... Eric Bray..... 215-848-5515
SECRETARY..... Mark Wannop..... 609-365-1776
TREASURER..... Don Arsenault..... 215-368-0446

Committees consists of:

TI-d-BITS Ralph Field..... 215-362-2534
Don Arsenault..... 215-368-0446
OPEN
Rice Hall
LIBRARY Ted Cheney..... 215-752-1458
Rich Mascara 215-441-4060
MEMBERSHIP ... OPEN
ASSISTANT TREASURER. Frank Passini
EDUCATION Barry Traver
Frank Passini
Ted Cheney
Tim Coyne
Carlo Angelico
EQUIPMENT Rice Hall
PROGRAM Dr. Eric Bray

REMEMBER to be considerate when calling any of the above people. Limit your calls to the early evening hours. (6pm to 9pm)

The opinions expressed herein are those of the individual authors are not necessarily those of the Philadelphia Area TI-99/4A Users' Group or its officers. Nor is the Philadelphia Area TI-99/4A Users' Group or any of its officers responsible for any damage, inconvenience, or loss which may result as a consequence of the use of any written material herein.

TI-d-Bits is published monthly by the Philadelphia Area TI-99/4A Users' Group, c/o Don Arsenault, 1290 Buttonwood Dr., Lansdale, PA 19446. All material herein may be reprinted freely by other non-profit User Groups, (unless otherwise stated), as long as proper credit is given to both source and author. Contributions are encouraged, but no payment is made. Editorial, advertising, and classified, copy MUST be in by the LAST day of the previous month. You can either mail your copy to: TI-d-Bits, c/o The Philadelphia Area TI-99/4A Users' Group, c/o Don Arsenault, 1290 Buttonwood Dr., Lansdale, PA 19446 or send it via modem by contacting Don Arsenault at (215)-368-0446. If your piece contains any diagrams, charts, or code, send a paper copy AT FINAL PUBLICATION SIZE.

Classified ads are printed in blocks. A block consists of 3 lines, 55 characters wide, or any increment of 3 lines. Classified advertising is accepted from members at NO CHARGE for a one block ad, per issue. Additional ads from members may be placed at cost of \$1.00 per block. Non-members may place classified ads at a cost of \$2.00 per block. All advertisements MUST be paid for in advance.

Commercial advertising is accepted for publication at the following rates:

Quarter page \$ 5.00
Half page \$ 8.00
Full page \$15.00

Commercial advertisements will be placed in the next available issue. All advertisements MUST be paid for in advance.

The editor of TI-d-Bits or the executive board of The Philadelphia area TI-99/4a Users' Group reserve the right to reject any material submitted for publication for any reasons.

The Philadelphia Area TI-99/4A Users' Group's program library is available to all active members at NO CHARGE for copying to your disk. A charge of \$2.00 per disk is made for club supplied disks for members. Non members may obtain copies of the library for a fee of \$5.00 per disk. A catalog of the library's contents is given to all new members upon request and updates will appear in this publication from time to time. To obtain material from the library, contact the librarian for the best procedure to obtain your requests.

THE PHILADELPHIA AREA TI-99/4A USERS' GROUP (MAY '90)

will also give you the number of bytes free.

After the PROMPT command, in this batch file, I have changed the current prompt to "F", because this is the device where the next batch file resides. The PAUSE command stops the execution until I press another key. When I press the next key, the batch file named BOOT will be executed. Let's now look at the file named BOOT:

```
ECHO OFF
CLS
F:
TYPE BOOT-M
PROMPT CHOICE?
```

This is a very short batch file. The first three command lines have already been explained elsewhere in earlier articles. The command TYPE BOOT-M is a variation of the TYPE command. The TYPE command will print a text file to the screen in the M-DOS mode. In this case, the text batch file BOOT-M, a menu file, is printed. The last command gives us a prompt with the word "CHOICE?". To see what the available choices are, we need to look at the file BOOT-M:

```
-----
          M DOS  MENU
-----
0.  RETURN TO M DOS
1.  TI-MODE (MOUSE MENU) GPL
2.  TI-MODE (SCHROEDER MENU) GPL
3.  PICTURE TRANSFER
4.  MY-ART
5.  HYPER-COPY
6.  RAMDOS
7.  NOT MY TERM
8.  MYCOLOR
-----
```

BOOT-M is a list of some of the files that I own that run directly out of M-DOS. These are the ones that I use most often, so I have places them on the first menu that I will see when I turn on my Genny. (With all of the new stuff that I have acquired, I now need to re-evaluate what files I want here. I also need to load a sub-menu or two just to deal with all that I now have available to me!)

Each of the choices on my M-DOS menu are numbered from

1 to 8. In order to run/load anyone of them, all that I need to do is to enter the appropriate number and press the ENTER key. Genny will then attempt to run a batch file with a name (number) from 1 to 8 that is located at the device designation that is determined by my present prompt letter. The third line BOOT has established the current prompt to be device designation "F", and this happens to be my first Horizon RAM disk. Now if I have batch files named (numbered) 1 to 8, residing at device "F" (DSK6), I am able to quickly and easily load/run them. Let us look at a few of these short batch files to see exactly what they do:

```
ECHO OFF
CLS
ECHO _____
VER
ECHO _____
PROMPT $_$d$b$b$t$_$_$u$y
MODE 80
F:
```

The above batch file is named "0", and this is the batch file that should return me to M-DOS mode. All of the commands, save one, have been discussed above. The command MODE 80 will merely toggle Genny back to an 80 column display and the prompt default is "F". Now let's look at the batch files named "1":

```
ECHO OFF
CLS
ECHO _____
ECHO _ LOADING MOUSE MENU & GPL INTERPRETER _
ECHO _____
F:
GPL EA
```

This batch file gives me a message regarding what file is currently being loaded. It then loads two files residing at device "F" It first loads my modified GPL with Randy Moore's mouse menu. It then automatically loads editor assembler into my GPL memory space. Let's now look at just two more batch files. This time, they will be files "3" and "4":

```
ECHO OFF
CLS
ECHO _____
ECHO _ LOADING PICTURE TRANSFER _
ECHO _____
ECHO _____
F:
H:\GRPX\PICT
```

The above file loads Picture Transfer from my hard disk. The M-DOS designation of my hard drive is device "H". The revised (back) slashes are necessary to separate

THE PHILADELPHIA AREA TI-99/4A USERS' GROUP (MAY '00)

my subdirectories. The directory is named GRPX, and the file to be loaded is named PICT.

ECHO OFF

CLS

ECHO

ECHO LOADING MY-ART FROM HRD #2 - DSK7.

ECHO

ECHO

ECHO

G:MYART

The above batch file loads the program MY-ART from my second Horizon RAM disk, DSK7. Using this syntax, I can load my M-DOS program from ANY HRD or floppy disk drive.

Well, I think that more or less rounds up my perusal of the subject of AUTOEXEC and batch files. Most of what I am presently using is based on stuff that Edward Hallett created and gave to the community more than a year ago. I am sharing with my readers the very same things that I am currently using to keep my Genny going, but I foresee making some changes in all of my AUTOEXEC/batch files. At this time, I project writing a sub-menu that will load TI mode programs directly from M-DOS using Barry Boons's fine EXEC program. I will be happy to share what I am able to get going, once I have it written and working. Please study what I have presented here. You may even want to type it all in and save it to disk using MY-WORD, TI WRITER, or FunnelWriter. After you have run it, change, adapt, and customize it to meet the needs of your own particular system. It really isn't hard to do. Now, onto something else:

SOMTHING NEW FROM JP SOFTWARE!!!

In my last article, I promised to write more about McCann Software's fantastic program The Printers Apprentice (M-DOS version). Unfortunately I have been unable to devote sufficient time to that project due to the passing of my mother-in-law this last month, but I hope to return to that subject in the next article or the one following it. Though my time has been very "messed up," I have been able to play with a couple of new programs which I picked up at the last Faire. They both come from another software company whose products I deeply respect. I am speaking about J.P. Software (formerly Genial Computerware). In my humble opinion, JP Software and Asgard Software are two of the most important and significant producers of software for our community. Here, I would like to express the fact that I am very favorably impressed with the amicable relationship that appears to exist between these two important 4A Community businesses. When I was at our last Chicago Faire, I described a piece of software that I was looking for to Mr. Chris Bobbitt. He unhesitatingly directed me to the JP Software table where I was able to find exactly what I

was looking for. I might also mention that the documentation for Asgard's Page Pro 99 contains a reference to the JP Software programs, Graphics Expander. (I love the professionalism of this business relationship.) I also find that these two companies also have something else in common: THEIR SOFTWARE OFFERING ARE VERY EASY TO USE! Let's now talk about Graphics Expander and use this as an example of what I am talking about.

I JUST LOVE MY GENNY'S GRAPHICS CAPABILITY!!!

Graphics Expander is a very useful program for anyone who has any serious interest in dealing with graphics on 4A or Genny. It allows you to rotate, invert, mirror, shrink, or expand any TI Artist or CSGD font or instance. It supports hard drive directory path loading, and it will catalog a disk drive and delete a file. This program is a very easy one to use. (I might also mention that small fonts really don't "shrink" very well. They just don't have enough pixels to play around with.) I find this program to be a real winner for me. One reason is that I have recently gotten very deeply involved in graphics using my Genny. As a result, I have acquired a great deal of the TIPS material that is out and available in our community. Having a hard disk and two Horizon RAM disks, I find that I am easily able to load and run long and difficult to load graphics programs. I therefore find myself often turning to TI Artist Plus, MY-ART, Macflix, Picture Transfer, Page Pro 99, and TPA with increasing regularity. In fact, I now find that tasks related to the use of graphics are now second to my word Processing related tasks. (This is interesting because I am NOT an artist!) I plan to devote an entire article to graphics on my Genny in the near future. The speed and ease with which Genny works only serves to further confirm my feeling that I don't NEED a mere "Klone" to do the jobs that I need to have done.

I JUST LOVE TO BROWSE!!!

The other program which I have also recently started to use, from JP Software, is the program BROWSE. In spite of my recent increase in graphics work, the great majority of my computer tasks are still word processing related. Therefore, BROWSE is a very useful program to me. This program allows me to catalog all of the DV/80 text files on a disk or hard drive directories. It also can write those files into its own buffer. Once a file is placed there, it can be either displayed on my CRT (in either 40 or 80 column format) or send to my printer for printing. Due to my work, I will often end up with a HRD, or a floppy disk, or a hard disk directory full of text files with names that are often somewhat similar. By using BROWSE, I am able to very easily see what I have. I can also print something out without having to load MY-WORD or Funnel Writer.

TIPS FROM THE TIGERCUB

#58.1

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213

I am still offering over 120 original programs at \$1 each, or on collection disks at \$5 each. The five Tips From The Tigercub disks are reduced to \$5 and the three Nuts & Bolts disks are now just \$10 each.

My catalog is available for \$1, deductible from your first order (specify TIGERCUB catalog).

TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 300 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an auto-loader by full program name on each disk. These are available as a copying service for just \$1.50 post-paid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or \$1, refundable for 11-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.

In Tips #55 I published a CHARSUB routine to convert character patterns into assembly source code, and in Tips #55 and #56 I published several routines to manipulate hex codes into

new character sets. Those patterns looked fine on my old TV, but when I demo'd them on a high-resolution monitor I could see too many missing pixels.

So I wrote this CHARFIX program which, when MERGED into a program and CALLED after any character redefinition is completed, will permit any normal or re-identified character to be viewed on screen and edited and will then write the hex codes of any range of printable characters into an assembly source file which can be assembled, loaded and linked to instantly change character sets.

This routine also reidentifies the common punctuation into the same character sets as the letters, as described in Tips #55. If you do not want this feature, delete lines 29001-29003.

```

29000 SUB CHARFIX
29001 DATA 32,33,34,44,46
29002 RESTORE 29001 :: FOR J
=1 TO 5 :: READ CH :: CALL C
HARPAT(CH,CH$):: CALL CHAR(J
+90,CH$):: CALL CHAR(J+122,C
H$):: NEXT J
29003 CALL CHARPAT(63,CH$)::
CALL CHAR(64,CH$):: CALL CH
AR(96,CH$)
29004 DISPLAY AT(1,1)ERASE A
LL:"1 2 3 4 5 6 7 8 9 0 ; "
:" @ A B C D E F G H I J
K L M" : " N O P Q R S T U
V W X Y Z [": "\ ] ^ _ a
b c d e f g h i j"
29005 DISPLAY AT(9,1):"k l m
n o p q r s t u v w x": " "
"y z { | } ~"
29006 CALL CHAR(128,"FF"&RPT
$("81",6)&RPT$("FF",9)&"FFFF
"&RPT$("C3",4)&"FFFF"):: CAL
L COLOR(13,2,16)
29007 CALL CHARVIEW
29008 SUBEND
29009 SUB CHARVIEW
29010 DISPLAY AT(13,14):"CTR
L V TO VIEW" :: DISPLAY AT(1

```

```

4.14):" " :: DISPLAY AT(15.1
4):"CTRL E TO EDIT" :: DISPL
AY AT(17,14):"CTRL S TO SAVE
"
29011 DISPLAY AT(19,14):" "
:: DISPLAY AT(20,14):" "
29012 CALL KEY(0,@,S):: IF S
=0 THEN 29012 ELSE IF @=150
THEN 29015 ELSE IF @=133 THE
N 29014 ELSE IF @=147 THEN 2
9013 ELSE 29012
29013 CALL DELSPRITE(#1):: C
ALL CHARSUB(HX$()):: DISPLAY
BEEP :: STOP
29014 CALL EDIT(K):: GOTO 29
010
29015 DISPLAY AT(24,1)BEEP:"
"
29016 DISPLAY AT(24,1):"PRES
S A KEY" :: CALL KEY(0,K,S):
: IF S<1 OR K<32 OR K>143 TH
EN 29016
29017 DISPLAY AT(24,1):"" ::
CALL CHARPAT(K,CH$)
29018 R=13 :: FOR J=1 TO 15
STEP 2
29019 H$=SEG$(CH$,J,1):: CAL
L HEX_BIN(H$,B$)
29020 H$=SEG$(CH$,J+1,1):: C
ALL HEX_BIN(H$,B$):: FOR L=
1 TO 8 :: C$=C$&CHR$(ASC(SEG
$(B$&B$,L,1))+80):: NEXT L
29021 DISPLAY AT(R,1):C$::
DISPLAY AT(R,10):SEG$(CH$,J.
2):: R=R+1 :: C$="" :: NEXT
J :: DISPLAY AT(22,1):CH$::
: GOTO 29012
29022 SUBEND
29023 SUB HEX_BIN(H$,B$):: H
X$="0123456789ABCDEF" :: BN$
-"0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
29024 FOR J=LEN(H$)TO 1 STEP
-1 :: X$=SEG$(H$,J,1)
29025 X=POS(HX$,X$,1)-1 :: T
$=SEG$(BN$,X$+1,4)&T$ :: NE
XT J :: B$=T$ :: T$="" :: SU
BEND
29026 SUB CHARSUB(HX$())
29027 DISPLAY AT(12,1)ERASE
ALL:"Source code filename?":
"DSK" :: ACCEPT AT(13,4)SIZE
(12)BEEP:F$ :: OPEN #1:"DCK"
&F$,OUTPUT
29028 DISPLAY AT(15,1):"LINK
ABLE program name?" :: ACCEP

```

```
T AT(16,1)SIZE(6):P$
29029 DISPLAY AT(18,1):"Rede
fine characters from ASCI
I to ASCII"
29030 ACCEPT AT(19.7)VALIDAT
E(DIGIT)SIZE(3):F
29031 ACCEPT AT(19,21)VALIDA
TE(DIGIT)SIZE(3):T
29032 PRINT #1:TAB(8);"DEF";
TAB(13);P$ :: PRINT #1:"VMBW
EQU >2024" :: PRINT #1:"
STATUS EQU >837C"
29033 NB=(T-F)*8 :: CALL DEC
_HEX(NB,H$):: A=768+F*8 :: C
ALL DEC_HEX(A,A$)
29034 FOR CH=F TO T :: IF CH
<144 THEN CALL CHARPAT(CH,CH
$)ELSE CH$=HX$(CH)
29035 IF FLAG=0 THEN PRINT #
1:"FONT";:: FLAG=1
29036 FOR J=1 TO 13 STEP 4 :
: M$=M$>"&SEG$(CH$,J,4)&",
" :: NEXT J :: M$=SEG$(M$,1,
23)&" *"&CHR$(CH)
29037 PRINT #1:TAL(0);"DATA
"&M$ :: M$="" :: NEXT CH
29038 PRINT #1:P$;TAB(8);"LI
R1, FONT" :: PRINT #1:TAB(
8);"LI R0,>"&A$ :: PRINT #
1:TAB(8);"LI R2,>"&H$
29039 PRINT #1:TAB(8);"BLWP
@VMBW":TAB(8);"CLR @STATUS"
:TAB(8);"RT":TAB(8);"END" ::
CLOSE #1
29040 SUBEND
29041 SUB DEC_HEX(D,H$)
29042 X$="0123456789ABCDEF"
:: A=D+65536*(D>32767)
29043 H$=SEG$(X$, (INT(A/4096
)AND 15)+1,1)&SEG$(X$, (INT(A
/256)AND 15)+1,1)&SEG$(X$, (I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
29044 SUB EDIT(CH)
29045 DISPLAY AT(13,14):"1 T
O TOGGLE" :: DISPLAY AT(14,1
5):"CURSOR" :: DISPLAY AT(15
,14):"E S D X TO MOVE" :: DI
SPLAY AT(17,14):"CTRL A TO A
PORT"
29046 DISPLAY AT(19,14):"CTR
L R TO" :: DISPLAY AT(20,15)
:"REIDENTIFY"
29047 R=13 :: C=3 :: X=128 :
: CALL SPRITE(#1,130,11,R*8-
7,C*8-7):: X$=CHR$(129)&CHR$(
146)
```

```
29048 CALL KEY(0,K,S):: IF S
<1 THEN 29048 ELSE ON POS("1
EeSsDdXx"&X$,CHR$(K),1)+1 GO
TO 29048,29049,29050,29050,2
9051,29051,29052,29052,29053
,29053,29055,29056
29049 X=X+1+(X=129)*2 :: GOT
O 29054
29050 R=R-1-(R=13):: GOTO 29
054
29051 C=C-1-(C=3):: GOTO 290
54
29052 C=C+1+(C=10):: GOTO 29
054
29053 R=R+1+(R=20)
29054 CALL LOCATE(#1,R*8-7,C
*8-7):: CALL HCHAR(R,C,X)::
GOTO 29048
29055 CALL DELSPRITE(#1):: S
UBEXIT
29056 FOR R=13 TO 20 :: FOR
C=3 TO 10 :: CALL GCHAR(R,C,
GH):: CALL LOCATE(#1,R*8-7,C
*8-7):: B$=B$&CHR$(GH-80)::
NEXT C
29057 CALL BIN_HEX(B$,H$)::
DISPLAY AT(R,10):H$:: B$=""
:: HEX$=HEX$&H$ :: NEXT R :
: DISPLAY AT(22,1):HEX$:: C
ALL CHAR(CH,HEX$):: HEX$=""
29058 CALL DELSPRITE(#1):: F
OR R=13 TO 20 :: DISPLAY AT(
R,14):"" :: NEXT R :: SUBEND
29059 SUB BIN_HEX(B$,H$):: H
X$="0123456789ABCDEF" :: BN$
="0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
29060 L=LEN(B$):: IF L/4>IN
T(L/4)THEN B$="0"&B$ :: GOTO
29060
29061 FOR J=L-3 TO 1 STEP -4
:: X$=SEG$(B$,J,4)
29062 X=(POS(BN$,X$,1)-1)/5
:: T$=SEG$(HX$,X+1,1)&T$ ::
NEXT J :: H$=T$ :: T$="" ::
SUBEND
```

I think that programs, at least non-commercial ones, should be open for anyone to modify for their own use. For that reason, I would not normally publish the following routine. However, I recently received a large number of programs,

originally in the IUG library, and found that the author's name had been erased from the title screen or RFM of every one of them. I know, because I already had many of the original versions, including some that I wrote myself.

Now, that is inexcusable. If a programmer is willing to share his work, he does deserve credit for it. And if people are going to play that dirty, maybe there is good reason for protecting programs.

So here is how to do it. Ken Woodcock wrote this ingenious routine and published it in the Tidewater newsletter. I have modified it so that it can be deleted after it has done its work. It is to be MERGED into any XBasic program(32k required) and RUN, and will change the line length byte of each line to zero, so that the program cannot be LISTed, although it can be loaded and run.

```
1 CALL INIT :: CALL PEEK(-31
95Z,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4
2 CALL PEEK(X,E,F,G,H):: ADD
=G*256+H-65536 :: J=J+1 :: I
F J<4 THEN 3 :: CALL LOAD(AD
D-1,0)
3 NEXT X :: STOP :: !@P-
```

Save that as FIX in MERGE format. Merge it into any program (RESequence first if it has line numbers less than 4) and RUN. Then type 1, FCIN X and FCIN 3 to delete line 1. Delete lines 2 and 3 in the same way. Then SAVE. Now try LISTing it and watch the fireworks.

Ken wrote an even more ingenious UNFIX routine to unprotect the program, but I'm not passing that on!

Now, suppose you have a party game program that you don't want the kids playing with. So, RESequence it to some odd number, such as RES 797. Put in a line just before that 796 STOP. Then merge in FIX, run it, and delete those first 3 lines.

I hope you remember what line number you resequenced it to start from, because now you can only run it by RUN 797 !

In Tips #57 I reported the discovery that printing to the disk from the TI-Writer Formatter, with the C option, really converted the carriage returns to trailing blank ASCII 32's, and I published a routine to strip them. I have found an easier way. First PF and C DSK... to convert the CRs to blanks. LF DSK... and SF DSK... to strip out those blanks, but that leaves the pestiferous tab line, so LF DSK... and PF DSK... again!

The first few disks of Tips #58 that I sent out had a poor version of this program. This is the corrected version. First key this in -

```

1 DISPLAY AT(12,1)ERASE ALL:
"SKIP INSTRUCTIONS? Y" :: AC
CEPT AT(12,20)SIZE(-1)VALIDA
TE("YNyn"):CQ$ :: IF CQ$="Y"
OR CQ$="y" THEN 8
2 DISPLAY AT(24,5)ERASE ALL:
"PRESS ANY KEY"
3 RESTORE 30721
4 REM
5 FOR J@=1 TO T@ :: READ @$
:: DISPLAY AT(J@,1):@$: " "
6 CALL KEY(0,K@,S@):: IF S@=
0 THEN 6
7 NEXT J@
8 DATA 0
9 RESTORE 8 :: READ N
10 REM
    
```

Save it by -
SAVE DSK1.D/MERGE,MERGE
Then key this in -

```

100 OPEN #1:"DSK1.D/MERGE".V
ARIABLE 163,INPUT :: OPEN #2
:"DSK1.D/MERGE2",VARIABLE 16
3,OUTPUT :: L=129 :: FOR J=1
TO 10
110 LINPUT #1:M$ :: PRINT #2
:CHR$(0)&CHR$(L+J)&CHR$(156)
&CHR$(253)&CHR$(200)&CHR$(1)
&"1"&CHR$(181)&CHR$(199)&CHR
$(LEN(M$))&M$&CHR$(0):: NEXT
J
120 CLOSE #1 :: PRINT #2:CHR
$(255)&CHR$(255):: CLOSE #2
    
```

Run it to convert D/MERGE into a merge format file D/MERGE2 on DSK1. Then key this in. Don't change line numbers.

```

100 CALL CLEAR :: OPEN #1:"D
SK1.@DATA",VARIABLE 163,OUTP
UT :: DEF L$(X)=CHR$(120)&CH
R$(X)
105 PRINT #1:L$(X)&CHR$(161)
&CHR$(200)&CHR$(6)&"@DUMMY"&
CHR$(0)
110 L=L+1 :: X=X+1 :: ACCEPT
AT(L,0):M$ :: IF L=24 THEN
CALL CLEAR :: L=0
120 IF M$<>"END" AND M$<>"en
d" THEN PRINT #1:L$(X)&CHR$(
147)&CHR$(199)&CHR$(LEN(M$))
&M$&CHR$(0):: GOTO 110
130 REM
140 PRINT #1:CHR$(0)&CHR$(4)
&"T@"&CHR$(190)&CHR$(200)&CH
R$(LEN(STR$(X-1)))&STR$(X-1)
&CHR$(0)
141 PRINT #1:L$(X)&CHR$(168)
&CHR$(0)
150 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1
    
```

Enter MERGE DSK1.D/MERGE2 to merge in that file. SAVE the program as DATAWRITER. Then RUN it and try it out by using it to write itself some instructions. Answer the prompts with -
DATAWRITER V1.2
by Jim Peterson

To be used to add instructions to programs.

Type the instructions and format them, centered or hyphenated or right-adjusted just as you want them to appear on screen, and enter each line. They will be written to a D/V163 file named @DATA. When finished, enter END.

Then enter NEW, then MERGE DSK1.@DATA, and RUN to see if everything is OK. If so, load the program needing instructions, make sure its lowest line number is more than 10 and the highest is less than 30721, and enter MERGE DSK1.@DATA.

And enter END, then OLD DSK1.DATAWRITER, then MERGE DSK1.@DATA.

PROGRAMS THAT WRITE PROGRAMS
PART 6
by Jim Peterson

The first five parts of this series were written long ago, but since then I have found a new method to write programs that really do write programs. I must give Karl Romstedt credit for this idea.

To illustrate this technique, I will use a program which writes an auto-loader to display a diskfull of programs by their complete name rather than the abbreviated filename. This is the LOAD program which I put on all my TI-PD disks.

First, we key in the part which will always be a part of the LOAD program. Do not change the line numbers because there is a reason for them, and leave that REM in line 11 because something

THE PHILADELPHIA AREA TI-99/4A USERS' GROUP (JUN '90)

else will be plugged in there later.

```

10 CALL CLEAR :: DIM M$(127)
:: CALL SCREEN(5):: FOR S=0
TO 14 :: CALL COLOR(S,2,8)::
NEXT S :: CALL PEEK(8198,A)
:: IF A<>170 THEN CALL INIT
11 REM
12 ON WARNING NEXT
13 X=X+1 :: READ M$(X):: IF
M$(X)<>"END" THEN 13
14 R=3 :: FOR J=1 TO X-1 ::
READ X$ :: DISPLAY AT(R,1):S
TR$(J);TAB(4);X$ :: R=R+1 ::
IF R<23 THEN 17
15 DISPLAY AT(24,1):"Choice?
or 0 to continue 0" :: ACCE
PT AT(24,26)VALIDATE(DIGIT)S
IZE(-3):N :: IF N>X-1 THEN 1
5
16 IF N<>0 THEN 19 :: R=3
17 NEXT J
18 DISPLAY AT(24,1):"Choice?
" :: ACCEPT AT(24,9)VALIDATE
(DIGIT):N :: IF N=0 OR N>X-1
THEN 18
19 CALL CHARSET :: CALL CLEA
R :: CALL SCREEN(8):: CALL P
EEK(-31952,A,B):: CALL PEEK(
A6+B-65534,A,B):: C=A6
+B-65534 :: A$="DSK1."M$(N)&
:: CALL LOAD(C,LEN(A$))
20 FOR J=1 TO LEN(A$):: CALL
LOAD(C+J,ASC(SEG$(A$,J,1)))
:: NEXT J :: CALL LOAD(C+J,0
):: GOTO 10000
10000 RUN "DSK1.1234567890"

```

Now, save that "source code" by SAVE DSK1.CAT/S,MERGE. Then key in this "assembler" which will convert the "source code" into an "object code."

```

100 OPEN #1:"DSK1.CAT/S",VAR
IADLE 163,INPUT
110 OPEN #2:"DSK1.CAT/O",VAR
IABLE 163,OUTPUT
120 FOR J=10 TO 21 :: LINPUT
#1:M$ :: PRINT #2:CHR$(0)&C
HR$(J)&CHR$(156)&CHR$(253)&C
HR$(200)&CHR$(1)&"2"CHR$(18&
1)&CHR$(199)&CHR$(LEN(M$))&M
$&CHR$(0):: NEXT J
130 PRINT #2:CHR$(255)&CHR$(

```

```

255):: CLOSE #1 :: CLOSE #2

```

Note what this routine does. It reads in each line of the tokenized CAT/S and prints it back out to CAT/O preceded by line numbers 10 to 21 in tokenized two-byte format followed by the tokens for PRINT #2, the tokens for a quoted string followed by the CAT/O record and the CHR\$(0) end-of-line indicator. Then it prints the double-255 end-of-file indicator and closes the files.

Now key in the CATWRITER program.

```

1 CALL CLEAR :: CALL TITLE(1
6,"CATWRITER"):: CALL CHAR(1
27,"3C429971A199423C"):: DIS
PLAY AT(2,10):"Version 1.4":
::TAB(8); Tigercub Softwar
e"
2 DISPLAY AT(15,1):"For free
":"distribution":"but no pri
ce or":"copying fee":"to be
charged." :: FOR D=1 TO 500
:: NEXT D :: CALL DELSPRITE(
ALL)
3 DISPLAY AT(2,3)ERASE ALL:"
TIGERCUB CATWRITER V.1.4":;
" Will read a disk directory
":"request an actual progra
m":"name for each program-ty
pe"
4 DISPLAY AT(7,1):"filename,
and create a merg-":"able 0
uickloader which dis-":"play
s full program names and":"r
uns a selected program."
5 DISPLAY AT(12,1):" Place d
isk to be cataloged":"in dri
ve 1 and press any key" :: C
ALL KEY(0,K,S):: IF S=0 THEN
5
9 OPEN #2:"DSK1.CAT",VARIABL
E 163,OUTPUT
100 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:N$,A,J,K :: LN=1000 :: FN=1
100
110 DISPLAY AT(12,1):"Disk n
ame?":N$ :: ACCEPT AT(14,1

```

```

)SIZE(-28):N$ :: LX$=STR$(14
-LEN(N$)/2):: LXLEN=LEN(LX$)
120 PR$=CHR$(0)&CHR$(11)&CHR
$(162)&CHR$(240)&CHR$(183)&C
HR$(200)&CHR$(1)&"1"&CHR$(17
9)&CHR$(200)&CHR$(LXLEN)&LX$
130 PR$=PR$CHR$(182)&CHR$(1&
81)&CHR$(199)&CHR$(LEN(N$))&
N$&CHR$(0)
140 PRINT #2:PR$
145 DISPLAY AT(23,1):"To omi
t a file, press Enter"
150 X=X+1 :: INPUT #1:P$,A,J
,B :: IF LEN(P$)=0 THEN 190
:: IF ABS(A)=5 OR ABS(A)=4 A
ND B=254 THEN 160 ELSE X=X-1
:: GOTO 150
160 DISPLAY AT(12,1):P$:"
PROGRAM NAME?" :: ACCEPT AT
(14,1)SIZE(25):F$ :: IF F$="
" THEN X=X-1 :: GOTO 150
170 PRINT #2:CHR$(INT(FN/256
))&CHR$(FN-256*INT(FN/256))&
CHR$(147)&CHR$(200)&CHR$(LEN
(F$))&F$&CHR$(0):: FN=FN+1
180 M$=M$&CHR$(200)&CHR$(LEN
(P$))&P$&CHR$(179):: IF X<11
THEN 150
190 IF M$="" THEN 210
200 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&SEG$(M$,1,LEN(M$)-
1)&CHR$(0):: LN=LN+1 :: M$="
" :: X=0 :: IF LEN(P$)<>0 TH
EN 150
210 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&CHR$(200)&CHR$(3)&
"END"&CHR$(0)
220 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #2
230 DISPLAY AT(8,1)ERASE ALL
:"Enter -":" NEW":;:" MERG
E DSK1.CAT":;:" DELETE ""DSK
1.CAT""":;:" SAVE DSK1.LOAD"
240 SUB TITLE(S,T$)
250 CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)
260 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1-S)+(J+1-S+13)+(J>14)*
3,J*(170/L),10+J*(200/L)):
NEXT J
270 SUBEND

```

Next, enter MERGE DSK1.CAT/O and that "object

code" will pop into place right after line 9. If you list it, it will look like a blown file, because most of the token codes are unprintable, but don't worry. Save the program as CAT- WRITER.

When you run the program, it will open an output MERGE format file called CAT and write those merged lines from CAT/O in MERGE format. Then it will open the disk you are cataloging, read the directory sector, and ask you for a disk name with the existing diskname as default. You can select any disk name you want to title the menu screen, up to 28 characters long. Line 110 computes the position to center the title, and lines 120-140 write to the CAT file a tokenized line 11 (overwriting that REM line) to display your title at the top of the screen.

Line 150 reads each filename from the disk directory, skipping over anything that is not a program (no one yet has been able to tell me how to distinguish an assembly image "program!"). For each filename, it will ask you for a complete program name. If you don't want a program on the menu (such as an XB program that is run from another program, or an image file), just press Enter. Otherwise the program name you select will be printed as DATA by line 170, in tokenized format in lines starting with 1100 (note the FN-1100 in line 100) and incremented by 1. Lines 180-200 assemble the filenames into DATA lines of up to ten names, and tokenize them in lines beginning with 1000.

When the last filename has been read, line 210 prints one last DATA item "END" to signal line 13 to stop reading, and then prints the double-255 end-of-file. Then you are given instructions to clear memory with NEW, merge in the CAT file, delete it because you don't need it any more, and save it back as LOAD.

When you list the LOAD program, you will find the original CAT/S restored in lines 10-19 and 1000, the line to display the title in line 11, the filenames in DATA lines starting with 1000 and the program names in DATA lines starting at 1100.

When you run the program, it will display the disk name, and read the filenames into an array. Then it will display the program names, numbered, on as many screens as necessary, and ask you to select a program by number. The corresponding filename by number is selected from the array, and lines 19-20 rewrite line 10000 to RUN that filename. List the LOAD program after you have used it to load something, and you will see that it has changed.

That algorithm in lines 19-20 was published in one of the earliest 99'ER magazines, in a letter by A. Kludge. It has been the basis for every XBasic menu loader, and has saved us uncounted thousands of hours. The author had asked me not to reveal his identity, but I think I can now tell you that "A. Kludge" was really the late Dr. Stefan-Romano, who passed away recently at the age of 57. He was a

brilliant man who did much for the TI world, at first as editor of the IUG library, and then through the Amnion library and Amnion Helpline. He was of great help to me on several occasions.

Some of you may have obtained from me a copy of CATWRITER which wrote GOSUB 21 in line 12, and CALL LOADs in lines 21-25 to change the cursor to my Tiger- cub emblem. If you have begun to have problems with the resulting LOAD program or with my previous Tigercub Menuloader which used the same CALL LOADs, I have finally found out the cause. When my Horizon RamDisk is on, any program containing those CALL LOADs will lock up the second time it is run!
