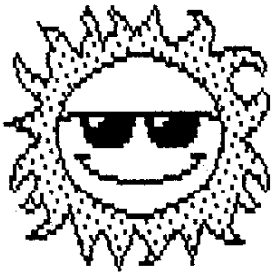
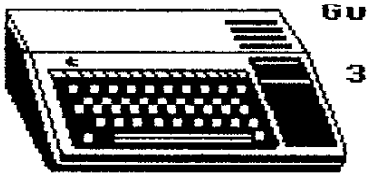
GUILFORD 99'ERS NEWSLETTER



SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER



GUILFORD 99'ERS UG
3202 CANTERBURY DR
GREENSBORO NC
27408

TO:

BOB CARMANY
1504 LARSON ST.
GREENSBORO, N.C. 27407

VOLUME 7 NUMBER 8 AUGUST 1990

Bob Carmany, Pres. (855-1538) . Emmett Hughes, Vice Pres. (584-5108)
Mack Jones, Sec/Treas (288-4280) Bill Woodruff, Pgm/Library (228-1892)
BBS: (919)621-2623 --ROS

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to:LF Jones, 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Ed.: 292-2035)

OUR NEXT MEETING

DATE: August 7, 1990 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Due to an illness in the family, last month's program was postponed. Tony Kleen has agreed to try again this month and give his demo of TI-BASE Vn 3.01. Come along and see the latest in the evolution toward the "perfect" data base.

CALL PEEK(PRES)

It isn't often that a product comes along that is the next best thing to sliced bread! I think that I have finally found something that is just that good! Back in April, I demo'ed my newly acquired Quest RAMdisk. The speed and ease of use convinced me that if one Quest was good, two would be better! With that in mind, I sent off a letter to my "mates" in Oz --Al Lawrence in particular -- to see if there might be one more of those thing floating around for a reasonable sum. A couple of weeks ago, a parcel arrived in the mail with the second Quest in it. A quick order to JDR Microdevices in California produced the requisite number of chips and --voila! -- a full meg of RAMdisk in the PE-Box.

I have Quest #1 partitioned as DSK4 and DSK5 and Quest #2 as DSK6 and DSK7. Just about everything imaginable is on the two of them and DSK7 is reserved as workspace for BBS downloads, text file storage and A/L assemblies. It even makes me wonder what 1 1/2 megs would be like --a third Quest . . .could be!

I may regret this next undertaking. At the urging, cajoling, and a downright "con job" by Tony McGovern and Ron Kleinschafer, I have undertaken learning Assembly Language. They convinced me that "it is the only way to program" and that is is "really very easy". It might be for the likes of those two but, frankly, I'm having a bit of a problem with it. I did manage to write a rather rudimentary program that displays a series of limericks on the screen. Even that simple program was "interesting" to work on! There were more than the usual bugs (wrong registers addressed), typos, and a bit of "cussin' and fussin'" before all was said and done. I did manage to make a start in A/L and the minor success of the program has encouraged me to think of loftier goals. Anyway, if anyone wants to bumble through A/L with me, give me a call and we will put together an informal Assembly Language class.

Due to an illness in Tony's family, the program was cancelled last month but will be re-done at this meeting. It was probably for the best since most of our "regulars" were either out of the country, sick themselves, or just plain forgot about the meeting. At any rate, with summer coming to an end, here's hoping that EVERYONE will come to this month's meeting for Tony's program. He always does an excellent job and I'm anxious to see this latest version of TI-BASE. I have never been able to get the stuff that I write for TI-BASE to behave properly and I have some questions for Tony before he even starts on the new stuff! This is running a bit longer than I had expected so 'til next month . . .

WHAT'S OUT THERE

With Tony doing a demo on TI-BASE, it is appropriate that we take a look at what is available for the TI in the realm of ---you guessed it --- databases.

Let's start with one of the simpler ones --CMINDEX. It is very simple but by the same token it is also easy to use. It gives the users a variety of output formats including mailing labels. On the other hand, there is no built-in sort routine and the field headings are fixed. It will let you catalog a collection of disks, maintain an address list, or a magazine library. The data is output in a D/V 80 (textfile) format and it can be manipulated with TI-WRITER with no modification at all. Since each record takes up one 80-column line, several thousand records could be stored on disk in a single file. For quick and easy database applications or something to learn about databases in general, it might be a good choice.

Next in this evolution is PR-BASE. This one has been through several versions and updates with the latest being Vn 2.1. It uses it's own format for storing records and, until this latest edition, the data was uncopiable with most disk manager programs. It is much more flexible than CMINDEX and supports custom graphics screens and other "creature comforts". There is a series of utilities available that allows for the actual re-sorting of data on your data disks. Sorting from within the program is a reorder of the index that is created when you initialize your file and start entering data. There are a number of single keypress commands and the documentation is well done. The limitations are 708 records per DSSD disk, though. For large applications this one might not be appropriate.

Now we come to TI-BASE! It is extremely versatile and flexible. The only trouble is that with all of this power comes a bit of complication. You must create your own command files to perform functions on the files themselves. It will store 16,000+ records per file and the files can be merged, manipulated and data exchanged between them. It is a superior database program but a bit difficult to use. In fact, for simple applications, one of the others might be a better choice. This one, though is about the best there is for the TI.

Of these three databases, the first two are fairware and the third is a commercial program. Take awhile to think about what you are going to be using your database for and then make a decision about which one suits your needs the best. Our UG library has both PR-BASE and CMINDEX in it and I'm sure Tony has the details about TI-BASE. It cost about $24 (and change) and if you are going to use just one database for varied applications, it is probably the one you need.

THE MISSING LINK

This review appeared in the April issue of the Hunter Valley 99'ers UG. The paragraphs have bee re-arranged but otherwise the text is the same as the original review.

I was fortunate enough to get a copy of THE MISSING LINK recently and everything that the author says about it in the review is accurate --and then some! It is easy to program with and the documentation is excellent. I would recommend it highly if you want some spectacular programming effects in XB. Maybe not the ultimate extension of XB, it certainly goes farther than anything that has been produced so far!

"This is a program of over 30 assembler subroutines that are available to the user in XB through CALL LINK's, much like the ENHANCED DISPLAY PACKAGE by PARAGON COMPUTING, but much more powerful and with many more features.

There are window drawing routines where the background colour in the window can be different from the screen colour. In fact, the main menu on the demo disk has five windows and each window has a different background colour with two colours of text.

The text commands seem very powerful. Through the use of bitmap graphics, text can start at any pixel row or column printing up to 32 rows and sixty columns. Also, you can have different size fonts on the screen at the same time. In fact, on one demonstration screen, there are six different window colours with six different text sizes in five different colours. Text input up to 255 bytes long except into a window and when text is input or displayed in a window you get automatic word wrap. Windows can be placed in graphic designs or graphics designs can be placed in windows.

The graphics are very impressive and fast with Cartesian graphics, points, lines, circles, and boxes plotted on the screen. TI-ARTIST and TI-ARTIST PLUS full colour pictures may be displayed and saved to and from the screen with the option of adding sprites and music to the picture. A full bitmapped graphics text screen dump is available."

[Ed note: The graphics part of the program is very much like LINES which came with the MINI-MEMORY cartridge except that you can limit the graphics to a window or display them outside of a window. The TI-ARTIST options open new horizons for XB programming.]

"Logo-like Turtle graphics can be generated in the multicoloured mode inside or outside windows with recursive operations allowed.

You can now have four more sprites to give a total of thirty-two and all thirty-two can move at one time plus precise control over whole groups of sprites which can be moved as one without drifting apart as they usually do in XB. Also whole groups can change colour at one time. Looks good for games programmers.

There is a routine called PAPERSAVER that shows on the screen how files prepared with Funnelweb or TI-Writer will look when they are printed, showing underlining and overstrike at least (they being the only things mentioned). It does this by showing the layout of the document on the left two-thirds of the screen with all of the words and letters as very small asterisks. The other third of the screen is taken up by two windows. The top window shows the file name of the document, the number of the page being displayed, and the line number of the page being displayed that the little pointer points to on the document. Under that is a list of the commands available. The other window, that is the bottom right window, actually shows the text (in green) for the line that the little pointer is pointing to, overstrike as black text and showing any underlining as well. I assume from the display that the text was to go through the formatter and that it reads all of the formatter commands. One thing about PAPERSAVER is that you don't have to go back to the formatter to print a document, PAPERSAVER has a hard copy command available.

The final thing on the demo disk is an analog clock with a sweep second hand. It may be all right for showing some of the capabilities but after the clock is drawn and you input the time it takes about twenty seconds to get it under way.

How did they do it? I don't know but the LOAD program is only about 12 lines long which shows up as thirty three sectors on the disk directory which means that it is about 8K long. It must have a heap of assembler code tacked on to the top of it. To have fitted all that into 8K seems incredible. The demo program is 97 sectors long which translates to about 24K and sizing after the LOAD program (with imbedded assembler code) shows 118840 bytes of stack left and 474 bytes of program space left. How much stack space gets eaten up with bit map manipulation a do not know. One thing they do not tell you in the pamphlets is how much room it actually leaves you for your programs. I realize that you still seem to have 24K in high memory but what about the VDP where all your strings are stored?"

[Ed note: The program is a well-written effort and appears easy to use with your XB programs. The only shortcoming is converting all of those graphic row and column values to dot row an column values for use in with THE MISSING LINK]

TONY'S CORNER

By Tony Kleen

Just a note to let you know that all is well with myself and my RAMdisk. My five-day capacitor apparently shorted out as I was inserting it into the PE box at the meeting. I've done this (removing and inserting) at my home several times and had no problem, but once I got away from home and in front of a crowd... Reminds me of a wayward child, kinda' like my own.

Anyway, I got it home, and after a two day cool-down period (mine, not the RAMdisk's), I reloaded the DSR and memory from an earlier backup; and things were going so great, I decided I would solve my problem of not having the RAMdisk's menu automatically presented upon power up. I noticed that with the club's TI disk controller, the RAMdisk had autobooted. My only difference was the CorComp. I proceeded to change the address in the RAMdisk to 1100 instead of the default 1000, then tried to power up. Then I tried again, again, and again. NOTHING!! Unless you call a dark black screen something! WHOOPS! I had better revert back to address 1000. No problem, ... , until power up. Powered up a couple more times. DARN!! That same dark screen was there. I wasn't able to destroy the RAMdisk at the meeting, but here at home, I was worried I might have.

When in deep doo-doo, call John McDevitt of RAVE99. Once again, John was right there to help. He told me what to do. Just remove the DSR 32K chip, reinstall, and reload. Memory was intact! The day was saved. I can't say enough good things about John's support of his RAVE products!!! Thanks again, John!.

Do you realize that TIBase 3.01 is out? If not, you don't read your MICROPENDIUM, nor come to our meetings. I received my latest copy last week. I had received 3.00 earlier, then returned my payment; and Dennis Faherty sent me the 3.01 without any questions. Here's another great guy supporting the TI!

As soon as I get this small blurb off to George, I plan to get into TIBase 3.01. These summer months sure cut into my TI pleasure. What with the pool, the ball games, the kids, not to mention the wife; it's past ten before I even get inside! Anyway, I plan to present Dennis's latest update to TIBase at our next meeting. See ya' there.

Tony Kleen

TEX-COMP NOTES

The other day I ordered some TI software from a mail-order house and instead of the familiar brown and white TI diskette jackets I received mine in a plain white jacket with the following note: "This program is out of production and at this time TI has not announced any plans for its continuation by TI or any other entity. Until such time that it may again become available. TEX-COMP is supplying a 'legal' copy using premium disks and cassettes. We report each sale to TI and will be paying them copyright royalties.....This service is provided as a convenience to our customers and to prevent their investment from being diminished." The accompanying documentation was evidently a photocopy of the TI original, complete with TI disclaimers, etc. PS: The diskette loaded without problems. While the paperstock of the documentation may not have been up to TI quality standards, who cares...the program worked and that is really what I paid my money for.

TEX-COMP offers a fairly full line of TI software, disk as well as cartridge software. From their catalogue it is impossible to tell which is a left-over TI original and what TEX-COMP "legal" copies....but who cares. By the way, TEX-COMP claims that they can supply a legal backup/replacement of any TI software in whatever medium if your original ever wears out. Mail in the original and a duplicating fee and they will supply the replacement.....Nice and comforting to know if you have an investment in one-of-kind cartridge software!

Allegedly Cor-Comp of California, up till now suppliers of third party memory and RS232 cards for the 99/4A, is working on a successor to the 99/4A called the 99/64. This computer is supposedly compatible with all the 99/4A hardware and software and will be priced in the $500 to $600 range. The basic unit will already include 64K of memory, the disk controller and a RS232 port. It will be interesting to see whether this is more than just a hot rumor!

BASIC TIPS

OR WHAT THE MANUAL DID NOT TELL YOU:

By Herman Geschwind

1. HOW TO DISABLE THE "FUNCTION-QUIT" HARDWARE RESET: TI Basic and Extended Basic has two ways to exit, one by typing in "BYE" which will properly close all files, or by pressing "Function=(QUIT)". The latter method really should not be used at all since files will not be closed and unpredictable things can happen if function quit is pressed while files are open. Unfortunately, many of us had the nasty experience of accidentally hitting "Function Quit" with the result that everything in memory was lost and files were scrambled. IF you have Extended Basic and 32K memory, the following will disable "Function Quit": CALL INIT::CALL LOAD(-31806,16). This can be typed in as a direct command, or could be the first line of an extended basic program.

2. HOW TO SPEED UP EXTENDED BASIC. While XB offers faster execution speed for some applications compared to console basic, XB can be speeded up even further by disabling sprite graphics (naturally this works only if the program does not use sprite graphics). The program statement is: CALL INIT::CALL LOAD(-31878,0). There are several different releases or versions of Extended Basic and the speed-up effect will be more pronounced with some versions than with others. 32K memory is required.

3. HOW TO RECOVER MEMORY IN TI BASIC/EXTENDED BASIC WITH DISK DRIVE ATTACHED. The TI operating system automatically sets aside memory to serve three concurrent open files. A minimum of 534 bytes of memory are taken up by general expansion overhead plus 518 more bytes for each of the three files opended up by default, or a total of just about 2K. If you know that you will have only one file open, key in the following DIRECT COMAND: CALL FILES(1)(Press ENTER) NEW (Press ENTER). This sequence will recover 1K of precious memory. Please note that this sequence can be keyed in as a command only and cannot be used as a program statement. Don't forget the NEW or results will be unpredictable. This prodecure can be used with both TI Basic or Extended Basic. With TI Basic and attached disk this is more essential than ever since TI Basic will only address 16K and you can ill afford to lose much of that.

XB PROGRAM

Here is a program listing that might help you remove some of the boredom when the weather turns colder in a couple of months. It is one of the old "peg jump" games. The object of the games is to jump one peg over another until you have only one (ha!) left. Keep track of the lowest number that you have left and go from there. Despite what you may think after playing the games a couple of times, it is possible go get down to a single peg.

```
100 REM
110 REM
120 REM
130 REM PEG JUMP
140 REM BASIC
150 REM
160 REM
170 REM
180 REM
190 CALL CLEAR
200 DEF MN=((V1>V2)*V2*(-1))
+((V1<V2)*V1*(-1))
210 DEF MX=((V1<V2)*V2*(-1))
+((V1>V2)*V1*(-1))
220 DIM RC(33,2),PLS$(8,4)
230 FOR I=2 TO 8
240 READ A,B
250 CALL COLOR(I,A,B)
260 NEXT I
270 DATA 7,7,14,14,5,5,8,1,8
,1,8,1,8,1
280 CALL SCREEN(1)
290 FOR I=0 TO 8
300 FOR J=1 TO 4
310 READ PLS$(I,J)
320 NEXT J
330 NEXT I
340 DATA JACKPOT!!,YOU'RE,TH
E,GREATEST,SUPER!!,BUT LAST,
PEG NOT,IN CENTER,EXPERT!!,A
LMOST,THE,JACKPOT
350 DATA PRO,PLAYER,STATUS,"
 ",AVERAGE,PLAYER,LEVEL," ",
NOT,QUITE,AVERAGE,YET
360 DATA GETTING,CLOSE,TO,AV
ERAGE,JACKPOT!!,IF I CAN,DO
IT SO,CAN YOU!!
370 CALL HCHAR(1,1,40,256)
380 CALL HCHAR(9,1,48,256)
390 CALL HCHAR(17,1,56,256)
400 A$="P E G  J U M P"
410 X=12
420 Y=10
430 GOSUB 2510
440 GOSUB 2410
450 CALL CLEAR
460 A$="STAND BY PLEASE"
470 X=12
480 Y=7
490 GOSUB 2510
500 RESTORE 2690
510 FOR I=1 TO 33
520 READ RC(I,1),RC(I,2)
530 NEXT I
540 FOR I=90 TO 132
550 READ A$
560 CALL CHAR(I,A$)
570 NEXT I
580 CALL CHAR(62,"0101010101
0101FF")
590 CALL CHAR(63,"8080808080
8080FF")
600 CALL CHAR(136,"1C1E7F7F3
E1C1C1C")
610 FOR I=1 TO 8
620 CALL COLOR(I,16,1)
630 NEXT I
640 FOR I=9 TO 13
650 CALL COLOR(I,2,16)
660 NEXT I
670 CALL COLOR(14,7,16)
680 CALL CLEAR
690 CALL SCREEN(5)
700 FOR I=1 TO 12
710 READ A,B,C,D
720 IF (I<5)+(I>8)THEN 750
730 CALL VCHAR(A,B,C,D)
740 GOTO 760
750 CALL HCHAR(A,B,C,D)
760 NEXT I
770 FOR I=12 TO 20
780 CALL VCHAR(2,I,99,21)
790 NEXT I
800 FOR I=8 TO 16
810 CALL HCHAR(I,6,99,21)
820 NEXT I
830 FOR I=3 TO 21 STEP 3
840 CALL HCHAR(I,14,97,5)
850 NEXT I
860 FOR I=7 TO 25 STEP 3
870 CALL VCHAR(10,I,98,5)
880 NEXT I
890 FOR I=13 TO 19 STEP 3
900 CALL VCHAR(4,I,98,17)
910 NEXT I
920 FOR I=9 TO 15 STEP 3
930 CALL HCHAR(I,8,97,17)
940 NEXT I
950 FOR I=1 TO 33
960 CALL HCHAR(RC(I,1)-1,RC(
I,2)-1,I+99)
970 NEXT I
980 A$="FROM:   TO:"
990 X=24
1000 Y=9
1010 GOSUB 2510
1020 RESTORE 2850
1030 FOR I=1 TO 4
1040 READ X,Y,A$
1050 GOSUB 2510
1060 NEXT I
1070 PL=32
1080 PR=0
1090 GOSUB 2300
1100 FOR I=1 TO 33
1110 CALL HCHAR(RC(I,1),RC(I
,2),136)
1120 NEXT I
1130 CALL VCHAR(12,16,96)
1140 A$=STR$(PL)
1150 CALL HCHAR(5,4,32,3)
1160 X=5
1170 Y=4
1180 GOSUB 2510
1190 A$=STR$(PR)
1200 X=5
1210 Y=23
1220 CALL HCHAR(5,23,32,3)
1230 GOSUB 2510
1240 IF (F88=1)*(PL=1)THEN 2
390
1250 IF F88=1 THEN 2370
1260 IF PL<8 THEN 1860
1270 CALL HCHAR(24,14,32,2)
1280 CALL HCHAR(24,20,32,2)
1290 T=0
1300 X=24
1310 Y=14
1320 GOSUB 2550
1330 F=VAL(A$)
1340 IF F=99 THEN 1070
1350 IF F<>88 THEN 1400
1360 F88=1
1370 RESTORE 2860
1380 CALL HCHAR(24,14,32,2)
1390 GOTO 1070
1400 IF (F<1)+(F>33)THEN 210
0
1410 X=24
1420 Y=20
1430 GOSUB 2550
1440 T=VAL(A$)
1450 IF MSG<>1 THEN 1470
1460 GOSUB 2300
1470 IF (T<1)+(T>33)THEN 210
0
1480 V1=RC(F,2)
1490 V2=RC(T,2)
1500 IF (RC(F,1)<>RC(T,1))+(
MN+6<>MX)THEN 1540
1510 MR=RC(F,1)
1520 MC=MN+3
1530 GOTO 1590
1540 V1=RC(F,1)
1550 V2=RC(T,1)
1560 IF (RC(F,2)<>RC(T,2))+(
MN+6<>MX)THEN 2100
1570 MR=MN+3
1580 MC=RC(T,2)
1590 CALL GCHAR(RC(F,1),RC(F
,2),CK)
1600 IF CK<>136 THEN 2100
1610 CALL GCHAR(MR,MC,CK)
1620 IF CK<>136 THEN 2100
1630 CALL GCHAR(RC(T,1),RC(T
,2),CK)
1640 IF CK<>96 THEN 2100
1650 CALL HCHAR(RC(F,1),RC(F
,2),96)
1660 IF RC(F,2)=RC(T,2)THEN
1740
1670 FOR I=RC(F,2)TO RC(T,2)
STEP SGN(RC(T,2)-RC(F,2))
1680 CALL GCHAR(RC(F,1),I,MC
R)
1690 CALL HCHAR(RC(F,1),I,13
6)
1700 CALL SOUND(-500,110*I,0
)
1710 CALL HCHAR(RC(F,1),I,MC
R)
1720 NEXT I
1730 GOTO 1800
1740 FOR I=RC(F,1)TO RC(T,1)
STEP SGN(RC(T,1)-RC(F,1))
1750 CALL GCHAR(I,RC(F,2),MC
R)
1760 CALL HCHAR(I,RC(F,2),13
6)
1770 CALL SOUND(-500,110*I,0
)
1780 CALL HCHAR(I,RC(F,2),MC
R)
1790 NEXT I
1800 CALL SOUND(-100,440,0,6
60,4,880,10)
1810 CALL HCHAR(RC(T,1),RC(T
,2),136)
1820 CALL HCHAR(MR,MC,96)
1830 PL=PL-1
1840 PR=PR+1
1850 GOTO 1140
1860 PLS=PL
1870 IF PLS>1 THEN 1910
1880 CALL GCHAR(RC(17,1),RC(
17,2),CK)
```

```
1890 IF CK<>136 THEN 1910
1900 PLS=0
1910 FOR I=200 TO 800 STEP 2
00
1920 CALL SOUND(100,I,3)
1930 NEXT I
1940 FOR I=18 TO 21
1950 CALL HCHAR(I,22,32,9)
1960 NEXT I
1970 FOR I=1 TO 4
1980 A$=PLS$(PLS,I)
1990 X=17+I
2000 Y=22
2010 GOSUB 2510
2020 NEXT I
2030 IF (F88<>1)*((PL<>1)+(C
K<>136))THEN 1270
2040 F88=0
2050 GOSUB 2410
2060 FOR I=18 TO 21
2070 CALL HCHAR(I,22,32,9)
2080 NEXT I
2090 GOTO 1070
2100 CALL SOUND(1500,110,0)
2110 A$="FR:"&STR$(F)
2120 X=18
2130 Y=2
2140 GOSUB 2510
2150 CALL HCHAR(19,2,32,6)
2160 IF T=0 THEN 2210
2170 A$="TO:"&STR$(T)
2180 X=19
2190 Y=2
2200 GOSUB 2510
2210 A$="ILLEGAL"
2220 X=20
2230 Y=2
2240 GOSUB 2510
2250 X=21
2260 A$="MOVE"
2270 GOSUB 2510
2280 MSG=1
2290 GOTO 1270
2300 FOR I=18 TO 21
2310 CALL HCHAR(I,2,32,7)
2320 IF (F<>88)*(F<>99)THEN
2340
2330 CALL HCHAR(I,22,32,9)
2340 NEXT I
2350 MSG=0
2360 RETURN
2370 READ F,T
2380 GOTO 1480
2390 PLS=8
2400 GOTO 1910
2410 CALL SOUND(400,131,3,16
5,3,330,1)
2420 CALL SOUND(400,196,3,26
2,3,330,1)
2430 CALL SOUND(400,147,3,37
0,1)
2440 CALL SOUND(200,220,3,29
4,3,370,1)
2450 CALL SOUND(200,330,1)
2460 CALL SOUND(400,196,3,37
0,1)
2470 CALL SOUND(1600,247,1,2
94,1,392,1)
2480 FOR I=1 TO 400
2490 NEXT I
2500 RETURN
2510 FOR Z=1 TO LEN(A$)
2520 CALL HCHAR(X,Y+Z-1,ASC(
SEG$(A$,Z,1)))
2530 NEXT Z
2540 RETURN
2550 A$=""
2560 CALL HCHAR(X,Y,42+ZZZ)
2570 ZZZ=ABS(ZZZ-1)
2580 CALL KEY(0,K,S)
2590 IF (A$="")+(K<>13)THEN
2620
2600 CALL SOUND(50,660,0)
2610 RETURN
2620 IF (S<>1)+(K<48)+(K>57)
THEN 2560
2630 CALL SOUND(50,660,0)
2640 CALL HCHAR(X,Y,K)
2650 A$=A$&CHR$(K)
2660 Y=Y+1
2670 IF LEN(A$)<2 THEN 2560
2680 RETURN
2690 DATA 3,13,3,16,3,19,6,1
3,6,16,6,19,9,7,9,10,9,13,9,
16,9,19,9,22,9,25
2700 DATA 12,7,12,10,12,13,1
2,16,12,19,12,22,12,25,15,7,
15,10,15,13,15,16,15,19,15,2
2,15,25
2710 DATA 18,13,18,16,18,19,
21,13,21,16,21,19
2720 DATA FF01010101010101,F
F8080808080808080,FF,8080808
080808080,FF00000000000000,0
101010101010101
2730 DATA 00183C7E7E3C18,000
0003C,000008080808,00,000000
0206020207
2740 DATA 0000000701020407,0
000000701070107,000000050507
0101,0000000704070107
2750 DATA 0000000404070507,0
000705010101,000000070507050
7,0000000705070101
2760 DATA 00000047C54545E7,0
0000042C64242E7,00000047C142
44E7
2770 DATA 00000047C14741E7,0
0000045C54741E1,00000047C447
41E7
2780 DATA 00000044C44745E7,0
0000047C54141E1,00000047C547
45E7,00000047C54741E1
2790 DATA 000000E7254585E7,0
00000E2264282E7,000000E72142
84E7,000000E72147B1E7
2800 DATA 000000E52547B1E1,0
00000E7244781E7,000000E42447
85E7
2810 DATA 000000E7254181E1,0
00000E7254785E7,000000E72547
81E1
2820 DATA 000000E725E525E7,0
00000E226E222E7,000000E721E2
24E7,000000E721E721E7
2830 DATA 1,12,94,9,7,8,94,2
1,17,6,92,21,23,12,92,9,8,5,
95,9,2,11,95,21,2,21,93,21,8
,27,93,9
2840 DATA 7,11,62,1,7,21,63,
1,17,11,90,1,17,21,91,1
2850 DATA 3,4,PEGS,4,4,LEFT,
3,23,PEGS,4,23,REMOVED
2860 DATA 15,17,28,16,21,23,
7,21,17,15,24,22,26,24,21,23
,8,22,29,17,33,25,22,24,31,3
3,18,30,4,16,33,25
2870 DATA 6,18,24,26,13,11,2
6,12,18,6,27,13,16,18,3,11,1
,3,18,6,13,11,10,12,3,11,12,
10,5,17
```

```
- TI-Base - Guilford - Tony Kleen -

SET TALK OFF
* ------------------------------ *
* TIB900703.BOOT/C               *
* ----------------------Vj------ *
* .............NOTE............. *
*                                *
*  When we DO BAK050, one RETURN's to *
* the 1st directive of BOOT!     *
* ------------------------------ *
*              I N I T I A L I Z E *
CLEAR
CLEAR LOCAL
LOCAL C1   C 1
LOCAL TLOC C 1
LOCAL 10C  C 2
LOCAL 16C  C 2
LOCAL Z    C 130
DO BAK002
*                 M A I N  M E N U *
WHILE (C1()"E")
  CLEAR
  WRITE 2,5 "TI-Base Report Writing"
  WRITE 6,15 "Output will be to the"
  WRITE 7,15 "-------------------------"
  WRITE 8,11 "P - Printer"
  WRITE 9,11 "D - Database (ARTICLE)"
  WRITE 11,11 "E - Exit"
  WRITE 13,11 "  - Selection?"
  SET RECNUM OFF
  SET HEADING OFF
  SET SPACES=0
  READCHAR 13,11 C1
  IF (C1="P").OR.(C1="D")
    REPLACE TLOC WITH C1
  ELSE
    IF (C1()"E")
      WRITE 23,1 ;
      "Invalid selection was attempted."
      REPLACE C1 WITH "E"
      DO BAK050
    ELSE
      SET SPACES=1
      RETURN
    ENDIF
  ENDIF
  IF (C1="D")
    CLEAR
    WRITE 2,4 "  TI-Base Report Writing"
    WRITE 7,16 "-------------------"
    WRITE 8,11 "Y - YES            "
    WRITE 9,11 "N - NO             "
    WRITE 13,13 "- Reset ARTICLE dbase?"
    READCHAR 13,11 C1
```

Well, I think I've had enough of a sabatical. I'm now ready to get back into my TI-Base presentations. I've got to tell you, that Dennis Faherty has certainly packed some punch into his latest release, 3.01. I've spent the last two weeks playing with it, and greatly improving my text presentation package. I told him during our last conversation how much I appreciated this latest enhancement. Atta' boy, Dennis!

What I will be presenting in the next several articles is a process that will take TIWriter files of 40 or 80 columns, combine them in various formats, and then write the final product to the printer or a database. If you've seen my articles before, you know that I like to put my 40-column command files to the left part of a page, and present an 80-column explanation of the command file to its immediate right. If you can't visualize what I'm saying, then just look at this page you're reading! NOW do you understand? ..Just being a little silly..

Anyway, the first thing one has to do is 'boot' the process. I bet you wonder what I call my 'boot' command file.. Good guess, BOOT. You'll notice the first thing is a SET TALK OFF, followed by a CLEAR screen directive. This is pretty standard for my boot'ing cf's. Also, I usually will SET the RECNUMbering OFF, and then SET the HEADING line printing OFF. The next segment of initialization is to define our local variables, along with their initial values. Two of the local variables are assigned in a separate cf, called BAK002. The reason for this?!? It's call planning ahead. Not everybody has the same printer command codes for 10 and 16 characters per inch. And, since I plan to package this process for anyone's use, I had better build some versatiliy into it. You'll later see a CUSTOMize command file that will step the user through customizing the printer codes, as well as RAMdisk customization.

I should also mention that little comment area. Part of my documentation is the date of the last update to this cf, the cf file's name, and what version number this is for the update date. As you can see, we are looking and the BOOT command file that is on its first version as of 90/07/03. That other note is just a reminder to me that upon return, I am always starting at SET TALK OFF. The reason for this?!? I'll explain it later.. Normally, when one calls another command file, the next directive in the calling cf is executed upon RETURN'ing from the called cf. With 3.01 of TIBase, you can 'trick' the operating system into always starting at the first directive by re'INSTALLing the BOOT cf into the 'install VDP' area. Actually, 'trick' probably isn't the proper term, 'destroying' the calling cf's pointer is more appropriate. Fortunately, TIBase will default to the 1st line when 'destruction' has occurred. I've just touched the surface on 'INSTALL VDP'. I've got some install cf's to describe later.

Now then, into the 'meat' of this cf. Three menues are presented to the user. The first is to determine where one wants ones printout directed. Your options are the PIO printer device, or to a database called ARTICLE. If one selects the database, then you're presented the second menu, where one decides if the database needs 'resetting'. The third menu is where one determines what presentation format is to be used. The first menu is always presented; the remaining two are dependent upon your selections.

In the 1st menu, if one enters selection (E)xit, the cf will SET SPACES=1 and then return to the .DOT command line. At this point, if you want to execute another printer presentation, you need only enter BOOT at the command line. This is a feature of 3.01's macro enhancement.

In the 1st menu, if one enters an invalid selection, you are informed of the error of you way, then control is passed to BAK050. In the BAK050 cf, you will notice that only the last two directives are executed for a C1="E" val What I do here is 're'INSTALL the BOOT cf into the VDP area, and then RETURN control back to BOOT. Since I have reinstalled the cf, and in effect destroyed

```
    IF (C1="Y")
      WRITE 23,2 "RESET'ing ARTICLE db."
      DO GAK017
    ENDIF
    REPLACE C1 WITH "D"
  ENDIF
* 
  WHILE (C1()"E")
    CLEAR
    WRITE 2,8 "TI-Base Report Writing"
    WRITE 08,10 "1 - 80       Column"
    WRITE 09,10 "2 - 80/40    Column"
    WRITE 10,10 "3 - 40/80    Column"
    WRITE 11,10 "4 - 40/40/40 Column"
    IF (TLOC="P")
      WRITE 12,10 "5 - ARTICLE Database"
    ENDIF
    WRITE 14,10 "E - EXIT"
    WRITE 16,10 "  - Selection"
    READCHAR 16,10 C1
    IF (C1("1").OR.(C1)"5")
      IF (C1()"E")
        WRITE 23,1 ;
        "Invalid selection attempted."
      ENDIF
    ELSE
      WRITE 23,1 ;
      "Begin selection's INSTALLation."
      DO GAK050
    ENDIF
  ENDWHILE
  REPLACE C1 WITH " "
ENDWHILE
RETURN
* ------------------------------------ *
*                                      *
* ------------------------------------ *
* TIB900630.GAK050/C                   *
* ----------------------V1------------ *
IF (C1="4")
  INSTALL LOAD GAK045
  DO GAK045
  INSTALL LOAD BOOT
  RETURN
ENDIF
*
IF ( C1="1")
  INSTALL LOAD GAK01B
  DO GAK01B
  INSTALL LOAD BOOT
  RETURN
ENDIF
*
IF (C1="2").OR.(C1="3")
  INSTALL LOAD GAK021
  DO GAK021
  INSTALL LOAD BOOT
  RETURN
ENDIF
*
IF (C1="5").AND.(TLOC="P")
  DO GAK025
  RETURN
ENDIF
INSTALL LOAD BOOT
RETURN
* ------------------------------------ *
*                                      *
* ------------------------------------ *
* TIB900630.GAK002/C                   *
* ----------------------V1------------ *
REPLACE 10C WITH ""
REPLACE 15C WITH ""
RETURN
* ------------------------------------ *
```

the previous copy of BOOT's return pointer; upon returning, the first line of the BOOT cf is executed. So, in essence, I'm returning to the main menu of the BOOT. In actuality, I'm reinitializing everything, but that really doesn't hurt anything! ...That's how the first menu works.

If you answer (D)atabase to the output question, you are presented a menu to determine if you want the database nullified. If you do, then a cf called GAK017 is executed. This cf simply copies my initial ARTICLE database to my working copy of the database. GAK017 will be discussed later. Upon completion of your selection, the formatting menu is presented. If you answer (P)rinter to the output question, you immediately are presented the formatting menu.

The final menu shows you your presentation formats. You can see that I currently allow a single 80 column, 80/40 and 40/80, and 40/40/40 columns. I have some other possibilities in mind! I won't present them until I have tested their validity! Also, I didn't mention the ARTICLE database selection. One is able to print (or re-print) what one has previously output'ed to the database. This is the way I distribute my articles. I first output my various formats to the database, and then let the newsletter editors just print them off. Something to be aware of, editors. V3.01 allows a 'report' option. I haven't yet researched this area, but have read that it allows 'headers' and 'footers'. If I can work it out, you'll be able to easily add your newsletter headers, and footer page numbers. What will they think of next???

On the third menu, if you enter an invalid selection, you are severely reprimanded with a statement mentioning this error. The third menu is continually presented until you enter a valid selection. Ahh, such punishment. If you (E)xit this menu, you are presented the main menu.

If you enter 1-)5 in the third menu, you'll notice that I go execute cf GAK050. GAK050 MUST reside on disk. This cf does all the INSTALLation of the formatting and boot'ing command files. If you're not familiar with 3.01, Dennis has essentially added a 'RAMdisk' area for your command file processing. He uses the 2.5k VDP area. All one has to do is 'add' his command files to this area, or 'load' a previously 'saved' set of command files that have been added to his VDP area. In GAK050, you'll notice that I am LOAD'ing a file (GAK045, BOOT, GAK01B, etc.) which in fact contain several command files. I used to execute my cf's on my RAMdisk. This is just as fast! Anyway, each selection will 'load' a set of command files, execute the 'master' cf, re-'load' the BOOT set of cf's, and then return to the BOOT cf's first directive.

If you like, I'll write a short blurb on INSTALL. Again, later!

That final cf is called GAK002. It is nothing more than an initialization of the printer codes for 10 and 16 characters per inch. I mentioned earlier that this is a 'customizing' cf.

Enough for one session. See ya' later.

The following is a direct, word-for-word reproduction of a recent IBM 'Service Support' announcement. (Really!)

ESD PRODUCT SERVICE SUPPORT
SUBJECT: NEW RETAIN TIP

Record number:	H013944
Device:	D/T8550
Model:	M
Hit Count:	UHC00000
Success Count:	USC0000
Publication code:	PC50
Tip key:	025
Date created:	089/02/14
Date last altered	089/02/15
Owning B.U.:	USA

Abstract: MOUSE BALLS NOW AVAILABLE AS FRU

Mouse balls are now available as a Field Replacement Unit (FRU). If a mouse fails to operate, or should perform erratically, it may be in need of a ball replacement. Because of the delicate nature of this procedure, replacement of mouse balls should be attempted by trained personnel only.

Before ordering, determine type of mouse balls required by examining the underside of each mouse. Domestic balls will be larger and harder than foreign balls. Ball removal procedures differ, depending upon manufacturer of the mouse. Foreign balls can be replaced using the pop-off method, and domestic balls replaced using the twist-off method. Mouse balls are not usually static sensitive, however excess handling can result in sudden discharge. Upon completion of ball replacement, the mouse may be used immediately.

It is recommended that each servicer have a pair of balls for maintaining optimum customer satisfaction and that any customer missing his balls should suspect local personnel of removing these necessary functional items.

P/N N33F8462 - Domestic Mouse Balls
P/N N33F8461 - Foreign Mouse Balls

SAS Keywords:
PSY2 8525SYSMISC 8530SYSMISC 8550SYSMISC
8560SYSMISC 8570SYSMISC 8580STSMISC

ESD PRODUCT SERVICE SUPPORT, BOCA RATON, FL.