QB MONITOR

QB-99'ERS U.G. NEWSLETTER

October 1988

The QB MONITOR is the Newsletter of the QB-99'ers User Group, is printed Sept. thru June and sent in exchange for other User Group Newsletters. Send Exchange Newsletter to Frank Cotty, Queensborough Community College, Bayside, NY 11364. Credit original sources.

The QB 99'ers meets the second Saturday of each month September through May, at Queensborough Community College, Bayside New York, room S225, at 2 P.M. Calendar at right shows dates

October 1988						
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

November 1988						
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

December 1988						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

GRIN & BEAR IT

"It's the hardening of the software."

Contents Page

QB MONITOR ~ QB-99'er NEWSLETTER

It grieves me to have to relate to you the passing of one of our respected longterm friends George Steffen. George was a member of the LA 99'ers Users Group. He wrote many articles for the Topics newsletter. He will be missed by all of us in the Home Computer family. In his memory we are reprinting an article he wrote in the Topics newsletter titled "What is on my disks?". May our memory of him live forever.

The TI computer has a way of attracting the nicest of people. Recently our group became aware of a nun, Sister Pat Taylor B.V.M. in Dubuque, Iowa who is using her newly found TI-99/4A to spread warmth to the senior citizens at the nursing home. All kinds of good things are being passed her way. Freeware programs, fairware programs (free from their authors), advice, help on hardware and software problems she is having and money are being sent to her in support of her work. P.S. she loves to write! One example of her unique talent is presented in this newsletter--"IT'S LATER THAN YOU THINK...." You can help too her address is:

Sister Pat Taylor B.V.M.
1050 Carmel Drive
Marian Hall #456
Dubuque, Iowa 52001

It is always a great pleasure to offer you the talent of our own members. I have that pleasure this month. Mike of the QB 99'ers continues his FORTHward HO! series this month----"STEP 3: MORE ON STRINGS, SIMPLE LOOPS"

Ed Machonis is a valuable programmer. His works, although not as intricate as Funnelweb, TELCO, or DM1000, are among the neatest feats of programming I have seen. His specialty is the TInygram. If you think not much can be done with a 48K computer.....he can change your mind with just a screen full of code. Ed presents FLEXI LABEL for our printing ease. Ed's inspiration is Mike Stanfill of the Dallas UG. Mike's programming is also represented in this newsletter..."TINYMIND" the game "codebreaker" in one screen of code!

Our series on RAM disks continues with PART VI from John Willforth of the West Penn UG. And concluding this newsletter a three page article written by Charles Good of the Lima Ohio Users Group. "The new features of Funnelweb V4.1" You must own a copy of Funnelweb loader it is the most significant software to arrive on the HOME COMPUTER scene ever!

QB-99'er NEWSLETTER
Reprinted From
Dallas 99 Interface
August 1988

FUNNYWEB FARMZ
by Mike Stanfill

Hiya, kiddies! It's Uncle Mikey and I'm back this month with more fun, games, and rip-offs of genuine commercial entertainment programs! YAYYYYYYY! Now, to find out who gets ripped-off this month we'll spin the big "Rip-off Wheel", so cross your fingers and hold on tight 'cause ..here...it...goessss! Okay, it's slowing down now. It's passed Parker Brothers. It's slipping by Wham-O and Atari and it's landing onnnnnnn.....INVICTA! Makers of that fabulous game of logic, MASTER MIND! So I guess we'll call this month's Tinygram "Tinymind". As appropriate a sobriquet as I've ever come across, but that's another story.

If you're one of those unlucky few who have never played Master Mind, then I suggest you check the back of your USER REFERENCE GUIDE and locate the game called Codebreaker. That's right, same game!

The basic playing mechanic of Tinymind is simple. The computer will choose a 4-digit number consisting of the numbers one through six. Your job is to guess at this code and solve it in six guesses or less and the computer will help you, in not one but two ways! For instance:

The computer has chosen the code number 6231. You INPUT 5261. Your lil' orphan will print beside it (In this example, you understand!) XXO. The two "X"'s tell you that you've placed the right numbers in the right spaces, namely the "2" and the "1". the "O" tells you that you've gotten another number right but have just put it in the wrong location, in this case the "6".

(I feel I should add a few things at this time. Type in, SAVE and RUN the game. You'll see a blinking cursor in the upper left corner of the screen. Type in a 4-digit number and ENTER it and the program will do the rest. This program will require nothing more than a console and XB. A monitor would be nice also. Now back to the rest of the story!)

As I've said before you only get 6 guesses but you'll usually find that adequate. If you want to make it easier by giving yourself a few more guesses then just change the 7 in line 9 to any number fewer than 20 (Or you'll over-write the end-game messages!).

Wanna make it harder? Give yourself fewer chances. Or if you want to be a REAL masochist, then change each FOR-TO in the program from 4 to 6. This will give you a six-digit number to solve and the possible combinations rise from a measly(?) 1,296 to 46,686!!!! Also, if you do this be sure to change the SIZE command in line 3 from 4 to 6 so that you'll be INPUTing the proper number of digits.

So have fun, and play nice!

```
1 !********TINYMIND********
   ********A TINYGRAM********
   ****BY MIKE STANFILL****
   **DALLAS TI USER GROUP**
   ************************

2 RANDOMIZE :: L=1 :: CALL C
LEAR :: A$="" :: FOR T=1 TO
4 :: B$(T)=CHR$(INT(RND*6)+4
9):: A$=A$&B$(T):: NEXT T

3 ACCEPT AT(L,1)SIZE(4)VALID
ATE("123456"):C$ :: FOR T=1
TO 4 :: D(T),E(T)=0 :: NEXT
T :: :: C=8 :: FOR T=1 TO 4

4 IF B$(T)=SEG$(C$,T,1)THEN
CALL HCHAR(L,C,88):: C=C+1 :
: D(T),E(T)=1 :: G$="."

5 NEXT T :: W$="YOU LOSE"

6 FOR T=1 TO 4 :: FOR G=1 TO
 4 :: IF D(G)OR E(T)THEN 8

7 IF SEG$(C$,T,1)=B$(G)THEN
CALL HCHAR(L,C,79):: C=C+1 :
: D(G)=1 :: G=4

8 NEXT G :: NEXT T :: L$="TH
E ANSWER IS " :: IF A$=C$ TH
EN W$="YOU WIN!" :: GOTO 10

9 L=L+1 :: IF L<7 THEN 3

10 DISPLAY AT(21,1):L$&A$&G$
:W$:"PRESS ENTER TO PLAY AGA
IN!" :: INPUT A$ :: GOTO 2
```

TUTORIAL

REPRINTED FROM **Topics - LA 99ers**

What Is On My Disks? by George F. Steffen

Many times I have been asked by members of our club to take a look at a disk because they have a program that will not run, or they do not know how to access a file. Often I find that they are trying to run someting that is not a program or that what they thought was a file was really a program.

The operating system which Texas Instruments put into their home computer is not the most versatile, it was designed for simplicity. In fact, it is so simple that many people operate the computer quite successfully without even knowing that there is such a thing as an operating system. All input and output is by way of "files" no matter whether it is to a printer, a modem, a disk, or whatever else might be connected. How the material in the file is organized is not of much importance when it goes to the printer, as long as it appears on the paper. However, if you have a disk and catalog it and wish to access a file which is listed in the catalog, it is much easier if you understand what it is.

If you examine the "TYPE" column of the disk catalog, you will see only five different entries. They are "Program","Dis/Var","Dis/Fix","Int/Var" and "Int/Fix". There is nothing sacred about the number of types nor the names. They were established by TI in its disk controller and reference manuals. There could be eight different types without using any more bits to keep track of them. As far as the operating system is concerned, all are files, even those that list as "Program." "Program" means a memory dump file; the information is stored exactly as it is in the computer memory. "Display" means the information is stored just as it would be on a printer, while "Internal" indicates that each variable is preceded by one byte giving the length of that variable. "Fixed" means that each record is given the full length and "Variable" records start with one byte giving the length of the record.

Program files probably were given that name because that is how Basic programs are stored. These files normally have four words at the beginning used to keep track of where the file belongs in memory and the rest of the file is just a copy of the computer memory. In addition to basic programs, these files may consist of memory image dumps of assembly language programs and files created by some TI modules (Personal Record Keeping and Statistics, for example). There are some assembly language programs also which store program segments and files in this manner. Therefore, if you give the command OLD and the program listed on the catalog does not load, it does not mean that there is anything wrong with either your computer or the disk. The only way to tell what is in that file is to examine the disk with Navarone's Disk Fixer, Thompson's Disk+Aid , Miller's Diagnostics or other similar disk utility.

Display files are usually text files created by a word processor of some type. They may also be the result of listing a program to a disk. Usually they are Fixed 80, but this can be altered if desired. The Editor of Editor/Assembler creates Dis/Var 80 and the Assembler creates Dis/Fix 80. If you save a basic program with the MERGE option, you will see it stored as Dis/Var 163. To load a program stored in this manner, you must enter the command NEW followed by the command MERGE (FILENAME). These files may be viewed by using TI Writer, Editor/Assembler or a print program like the one I published in the Newsletter a couple of months ago.

Internal files usually have been created by operating programs. However, if you are using Extended Basic and Memory Expansion, a program too long to fit in VDP memory will be stored as Int/Var 254. If the program is not over 49 sectors on a disk, you may be able to convert it to a program file by using the commands CALL FILES(1), NEW, OLD (FILENAME) and SAVE (FILENAME).

When files are stored on disks, the file information block tells which type it is. However, there is no such header on tape, so you must be sure you know the type and arrange for input using the same type. If you use the wrong type, you may see some strange results.

QB MONITOR ~ QB-99'er NEWSLETTER

Reprinted From Cedar Valley 99er U.G. Newsletter

We can all relate to the following, from Sister Pat Taylor.

IT'S LATER THAN YOU THINK....

If you remember ages ago when a boot referred to footwear, not a program coming on a computer...

Or the days when a bulletin board was simple school equipment and not electronically sent messages...

Or times when we spoke of animation in reference to a person's enlivened spirit, not a screen imitating life...

Or those eons ago when conversions were political or religious, not binary or Ascii...

Or when utilities were electrical bills, not terrific helpers to aid a floundering computer novice...

Or when default referred a failure to meet an obligation instead of a comfortable choice made for you...

Or another century when alpha and omega were companions in speech instead of alpha and lock...

Or a time in the past when dip referred to a condiment and switches were a tool for chastisement ...

When strings were musical instruments, not a sequence of characters

When terminals were bus and train depots , not input, output devices...

When a buffer shined one's floors, and was not a temporary storage device...

When bugs were insects, not an error in a computer program...

Or when monitors watcheds school halls and lavatories instead of you watching a screen performance...

When emulation was striving to excel, not computer imitating computer...

Or a bus was transportation, not a circuit...

When call meant to phone, not to branch to a subroutine...

When a dump took care of waste materials, and was not a lovely print-out...

When address was where you lived, not an operand portion of a computer instruction...

When memory belonged to people, not computers...

Those belonged to another day, another year, and even at times another century!!! Is it later than we thought, or is it simply a question of the past not quite yet converted to the future!!! Could it be only a year ago when computerese first overwhelmed me!!

WANTED: I'm trying to locate an AC line stabilizing transformer or a constant voltage transformer for Sister Pat Taylor. Sola is a big name in these. It needs to be rated about 500 watts. The voltage level in her room fluctuates wildly, mainly on the low side. This is due to age and additional air conditioning loads in her building. Her computer system won't even boot at times. I will accept and deliver any contribution of a transformer. Gary Bishop, 377-9574 (H); 395-8267 (O).

FORTHward HO! STEP 3:
MORE ON STRINGS, SIMPLE LOOPS

More on FORTH strings, simple loops, in part 3 of our trek through the world of TI-FORTH.

by MikeH1 QB99'ers

Let's start off by taking a further look at creating and using string data in TI-FORTH. In our last discussion, we looked at one way strings could be created, stored and retrieved in TI- FORTH. We directly alloted space for the string, then using the FORTH word EXPECT, we entered the string into the alloted space.
When we wanted to retrieve the string, we used the FORTH word TYPE along with the number of letters from the string we wanted TYPED.
As is usually the case, there is another way to handle strings in FORTH This new way takes care of always having to specify the length of the string in use. This method also will introduce two new FORTH words to us...

-CREATE

(a variation on the TI-FORTH word CREATE that allows compatibility with the FORTH 83 STANDARD) is used to create a string literal, that can be retrieved with the familiar FORTH word TYPE, and the new FORTH word COUNT. Lets see just how this works...

First we have to create the string literal. Supposed we wanted to create the string variable MSG. This would be done with the FORTH statement

```
-CREATE MSG 42  STRING THIS IS A MESSAGE*
```

-CREATE MSG creates the variable MSG. 42 is the ASCII code for the string deliminator, or the character that specifies the end of the string. STRING identifies the variable as a string type variable.
THIS IS A MESSAGE is the text, or string of characters placed in the string variable MSG. And the asterisk * identifies the end of the string.

O.K. now that we have MSG, what can we do with it???

Well, for starters, lets see how we can retrieve the data, or string at MSG..

Remember the FORTH word TYPE.. well we use TYPE with the new word COUNT in this manner to get the string MSG to print out..

```
MSG COUNT TYPE
```

As you can see it is no longer necessary to specify the number of characters in the string, the word COUNT takes care of that for us.

O.K. lets move on to simple loops in FORTH.

LOOPS are very handy elements of any programming language. The ability to do repetitive tasks over and over again is a powerful aspect of computer programs, and this is made possible by the LOOP.

In TI-BASIC we are familiar with the FOR-NEXT LOOP. In TI- FORTH we have something that is very similar to that. It is called the DO LOOP. There is also another type of loop called the BEGIN UNTIL LOOP. Lets start off with the DO LOOP, and see how it works. The following FORTH statement will serve to illustrate a simple DO LOOP...

```
: TEST  10  0  DO  CR ." HELLO THERE " LOOP ;
```

As you can see we defined a new FORTH word TEST in which to run our loop. That is one of the prerequisites for using a LOOP in FORTH. Any loop must be used within a definition.

The first two numbers 10 and 0 define the upper limit of the loop and its starting point respectively. This means that our loop will run up to ten times starting from 0.
The word DO identifies the actual loop sequence. Anything between DO and LOOP in the statement will be carried out the specified number of times.

By the way, if you didn't notice, we entered our first program as it were. TEST is a FORTH DEFINITION, or a new FORTH word. By building a DICTIONARY of FORTH words you can build what we have become used to thinking of as a PROGRAM. We will take a closer look at LOOPS in our next FORTHward trek, and we will go onto the topic of CONDITIONAL BRANCHING, a fancy way of saying decision making computer style.

Till next time....

```
* : -CREATE <BUILDS DOES> ;
```

LET'S TALK RAM DISKS PART VI

By John F. Willforth

Well it's May 1988, and I've been writing about RAM DISKS for six months now. I've only lightly covered the "BIG FOUR", HORIZON, HRD+, MYARC, and CORCOMP. Each of them is available now, and most if not all the bugs have been ironed out. I thought that if I stretched it out long enough, that DATABIOTICS would have their unit out, running, some deliveries under their belt, and be picking up steam in the RAM DISK market for the TI. Well there is STEAM!

I'm not going to get into all the details, (it's hard to know what is true) but in order to give those of you who are chomping at the bit for the "ADVERTISED" GRAND RAM, I'll review an article by a guy who has two of these rascals, Ken Hamai of the Users Group of Orange County, CA. He issued many of his own opinions below, and where this was done, I'll put " " around the text.

Ken says that he's been beta testing a pair of the GRAND RAMS from DATABIOTICS, and "they have really gotten it together with the software. It has been unfortunate that this product has had so many delays in manufacture, but in a way, looking back over the last 6 months, I have to admit that the software six months ago was terrible compared to what it is now." Ken then compares the Geneve, and infers to the long incubation, nativity, infancy, and it's nearing puberty even now, as a justification for the GRAND RAM'S long time in development. I personally don't find that the comparison is there. It takes more to develop a machine that emulates an older technology, and maintain compatibility with all peripherals that plugs into the same SYSTEM BUS, as well as is able to keep up with the best of what is being produced by the monied personal/home computer manufacturers of today than it is to follow on the coat-tails of people like LOU PHILLIPS, RON GRIES, JOHN CLULOW, DAVE ROMER, and the folks at CORCOMP.

The West Penn 99'ers attempted to purchase two RAM DISKS from DATABIOTICS as long ago as TWO years. We were buying them as part of the NATIONAL ADVISORY PANEL (NAP), which was set up by DATABIOTICS to get software/hardware into the hands of key people in the TI family for the purpose of reviewing for the mutual benefit of DATABIOTICS and the Users Groups. This was to be the GRAND RAM yes 2+ years ago not six months ago. REMEMBER? We did get our money back about a year ago after being without it over 11 months. We just live right!

Enough, I think we need all the support we can muster, and we in turn must support those who take the plunge and invest thousands in an orphaned computer. We have to as editors of newsletters, include the complete picture so that you get the most for your dollar and are informed about what is going on in the marketplace.

Ken reports that you can use the GRAND RAM as both a RAM DISK and as a PRINT SPOOLER, with the sizing being done by a CONFIGURE program not unlike that available for the HRD, HRD+, but having of course the added ability to configure the CLOCK, the PRINT SPOOLER as well as of course the RAM DISK. Ken says that you can have up to seven of these GRAND RAMS in a PEB, and "use them all". He set his 512K (max. unit size) card up as 1-DSDD, 1-SSSD, and a 50K spooler. A 50K spooler should be more than enough to print out most doc files, and return the TI CPU to you for further processing in less than two minutes. The printer may start to bother you after a half an hour of printing while your playing VIDEO CHESS.

The drives can be designated 1 thru Z, and this should come out to 35 logical units and if that isn't enough, you better look to a WINCHESTER DRIVE.

The GRAND RAM has "HOT KEYS", which means you don't have call loads. "You just punch the keys and WHAM! The program you want is up and running!" This is because the unit has a built-in interrupt timer chip, and each card has 14 HOT KEYS. These keys do such things as turning your clock on/off, print spooler on/off, power up program on/off, as well as disabling interrupts, and seven keys (programmable) for program loading. Remember each GRAND RAM in the PEB has these 14 keys.

The unit comes with a modified "MENU" program from John Johnson, updated version of 4A TALK, and the word processor is modified (which WP, I don't know) to run on it. Disk Master I (from Databiotics) is provided but at this time cannot access drives 7 thru Z. John Birdwell's Disk Utilities 4/Gr supports this unit according to Ken, and "does it all!" I believe that if I were to get a GRAND RAM, I would write to John Birdwell at the same time I wrote the check for the hardware.

Speaking of hardware, the price is up a bunch since you read about that $99. price last summer for a 64K version ($150. now). Be careful about buying this unit right now. Ask the guys in Dallas, Texas. It's really expensive when you don't get the hardware!

THE NEW FEATURES OF FUNNELWEB v4.1

by Charles Good
Lima Ohio User Group

FUNNELWEB is probably the most significant software ever for the 99/4a. After booting FUNNELWEB v4.1 from XBASIC (you can boot FWB from any assembly language loader, but the XBASIC module is the best way) you can do all of the following without changing modules:

1. With a single keypress you can load from a selection of user created menus almost any software ever written for the 99/4A. If the software you want to load isn't configured into one of your user created software menus, you can call up a disk directory anywhere within FWB, mark the file name of software seen in the directory, and then load that software.

2. Do work processing with a much improved version of TI-Writer.

3. Create assembly source code and then assemble it as you would with the E/A module.

4. Manage disks with a modified version of DM1000 which is supplied with the FWB package. Pre-configured menu entry points for other common disk managers are also provided.

5. View and edit disk sectors with a modified version or DISK PATCH, also sometimes known as DISK0.

This review will describe the changes and additions in v4.1 as compared to v4.0. Although this description is based on the May 30, 1988 release which says "Memorial Day" on the XBASIC title screen, the review should be valid for all subsequent releases of v4.1.

Enhanced CENTRAL MENU capabilities.

Each central menu now has 8 items, and items 4-7 are completely configurable to load any kind of assembly language file. This includes autostarting D/F80 source code and assembly PROGRAM files. In previous versions of FWB the central menus could only load PROGRAM files and only a limited number of central menu slots were configurable.

The TI-Writer menu reads as follows:

1 EDITOR
2 FORMATTER
3 DISK UTILITIES
4 MODEM
5 DATA BASE
6 DM1000
7 DSKU
8 USER LIST

As noted above, items 4-7 can be configured to suit the user. MODEM is an entry point for terminal emulation software such as FAST TERM or TELCO. DSKU refers to John Birdwell's "DISK UTILTITIES." This fairware disk manager/sector editor is so good that some former users of DM1000, myself included, have switched to DSKU for most disk management uses. DSKU is not provided as part of the FWB package, but can be obtained directly from John Birdwell or from most user group libraries.

Item 3 in the above TIW central menu leads to a specially created user list menu in which disk management software is grouped together. The DISK UTILITIES menu reads as follows:

1 DM1000
2 DSKU
3 MYARC DM
4 DPATCH
5 SCREAMER
6 TRACKER
7 ARCHIVER
8 CONFIGURE
9 <CTR ROM>

DPATCH is the modified sector editor DISK0 which is provided as part of the FWB package. SCREAMER is a good entry point for an ultra fast whole disk copier such as REDISKIT or TURBO COPY. TRACKER can be used to load one of the various "copies anything including protected disks" track copiers. Will McGovern, one of the FWB authors, has written a fairware track copier called TRACKER that is one of the few (maybe the only) that works with a Myarc disk controller. Send him a few bucks in Australia and he will send it to you, or look in your user group library. ARCHIVER will load the latest version of Barry Boone's archiving/compressing program. This archiving software is not part of the FWB package. CONFIGURE boots the FWB configuration files CF/CG. Items 1-8 in the above DISK UTILITIES user list menu can be altered with CF/CG to boot any assembly D/F80 (autostarting or not) or PROGRAM files.

The Edit/Assm central looks this way as configured on the FWB distribution disk:

1 EDITOR
2 ASSEMBLER
3 LOADERS
4 C-COMPILER
5 DISK PATCH
6 LINEHUNTER
7 ..
8 RESET

Item 4 loads the latest v4 release of c99. LOADERS, unchanged from FWB v4.0, leads to a menu for loading assembly D/F80 or PROGRAM files that aren't already configured into one of the FWB user lists. LINEHUNTER is new to v4.1. It is an assembly programming utility that prints on the screen any specified line of assembly D/V80 source code. You can also type the name of a label, and LINEHUNTER will display lines that have that label.

THE CONFIGURATION PROGRAM, FILES CF/CG:

This has been totally redone for v4.1 and MUST be used to do any configuring of the various user lists. It is no longer possible to directly edit FWB's XBASIC LOAD program to alter the XBASIC user list because there is very little XBASIC code in LOAD. There are only a few XBASIC line numbers in LOAD and the rest of LOAD is all in assembly.

CONFIGURE is much easier to use in v4.1 than it was in v4.0. CF/CG has a tree structure which allows you to quickly get to any part of the configuration without redoing the entire configuration process. The configuration program is very professional looking with sound effects, overlaping menu windows that pop into view, and help screens that are available at various points in the configuration process by pressing "?". Obviously much effort went into the preparation of the new v4.1 configuration files. The authors note that CF/CG was condensed from over 500 sectors of source code.

Basically what you do is load a configuration data file, alter the configuration, resave the altered data file to disk, and then while it is still in memory install the new configuration data into the FWB LOAD and UTIL1 files. If you later obtain a more recent release of FWB v4.1 you can configure the more recent release simply by loading your old configuration data file and installing this data into the newer LOAD and UTIL1 files. This is REALLY EASY! I had access to a pre-release beta testing edition of FWB v4.1 and was able to use the beta testing edition's configuration data file to configure my "Memorial Day" v4.1 in about 30 seconds. Hopefully the FWB authors will maintain this configuration data file system in all future upgrades of FWB and allow this easy transfer of v4.1 configuration information into all future FWB updates. Unfortunately, configuration information from v4.0 cannot be transferred directly to v4.1.

The important universal keys to remember in the configuration routine are <ENTER> to advance to the next window, and BACK to return to the previous part of the program (often the previous window). AID will get you a disk directory from most places in the program, and "?" brings up the help screens. When a window is displayed, you press the first letter of the text line in the window to perform the function indicated. If the window says

```
 _____
!Load!
!Edit!
!Save!
 ------
```

then you press "L", "E", or "S". Sometimes "N" and "B" are used to move the cursor up and down within a window when the functions "Next" and "Back" are displayed in the previous window. When more than one window is visible at the same time, the active window is indicated by a fat (2 pixel wide) border. The borders of the windows get alternately fat and skinny depending upon which is the currently active window.

The early windows in the configuration process are as follows:

```
 ---------------------------------------
 ! Sysinfo        Quit        Install !   FIRST
 ---------------------------------------   WINDOW

 _____   _________     ________________
!Load!   !Loading!     !Boot Tracking OFF!
!Edit!   !Devices!     !TI-Writer side 1 !   LOADING
!Save!   !Colors !     !Edit/Assm side 1 !   WINDOW
 ------   !Menu   !     !Working Drive  2 !
SECOND    !XB List!     ------------------
WINDOW    !UL List!
          ---------      ____________
            THIRD       !Edtr Printer!
            WINDOW      !Fatr Printer!   DEVICES
                        !Object File !   WINDOW
                        !Work File   !
                        !Program     !
                        -------------

                        _____
                        !Edit!
                        !Next!   COLOR
                        !Back!   WINDOW
                        !Xchg!
                        !Redo!
                        !View!
                        ------

                        ________________
                        !TI-Writer side!   MENU
                        !Edit/Assm side!   WINDOW
                        ----------------
```

You start out by pressing S(ysinfo) to display the second window, and then press L(oad) to load the configuration data file, following the prompts for loading. The name of this file on the FWB distribution disk is SYSCON, but you can use any name. You can create different FWB configurations on different configuration data files each with a different file name. After loading the configuration file, you press E(dit) change the the configuration data and display the third window. From here you bring up fourth series of windows where much of the configuration actually occurs.

LOADING WINDOW: Boot Tracking toggles between ON and OFF by pressing "B". Usually it is left ON unless FWB is installed on a ramdisk. In that case, the authors suggest it is sometimes best to leave boot tracking OFF. If the files loaded by FWB's central menus are in different drives (or ramdisks) these drive numbers are specified next to "TI-Writer side" and "Edit/Assm side". The "Working Drive" is the default drive number that appears after "DSK" if the mailbox workfile name is empty when you LF from the editor or use the Formatter.

DEVICES WINDOW: Printer names are self explanatory. "Object File" name is the default that appears on the screen next to DSKx. when you select LOADERS from the Edit/Assm central menu and try to load a D/F80 assembly file. I have "Object File" name configured as "DF/80FILE" to remind me that only this type of file can be loaded from certain parts of LOADERS. "Work File" is the default file name used the first time you LF. It is best to leave this blank as it is on the distribution disk. If "Work File" is left blank, you can exit FWB, go through the title screen and do something in BASIC without turning off the PE box, return to FWB and find the previous workfile name still there when you LF. "Program" is the default displayed on the screen when you use the LOADERS menu to load assembly PROGRAM files. The ability to set "Object File" and "Program" defaults is a feature new to FWB v4.1.

COLOR WINDOW: Here you can edit and view your choice of 10 color combinations. You can alternate between two sets of such combinations. The one listed at the top of the list is the combination that appears first when you boot FWB.

MENU WINDOW: This choice allows you to configure items 4-7 in each of the central menus. You can configure autostarting D/F80 assembly files or assembly PROGRAM files. Each file name is limited to two characters and should either be on the boot disk in the either one of the two central menu drive numbers specified above in the LOADING MENU. Press BACK to exit this part of the configuration process.

XB LIST AND UL LIST CONFIGURATION: These are both done in a similar manner. First F(etch) the list by pressing "F". Then press N(ext) or B(ack) to select the item to be configured and press E(dit) to change that item. Press <ENTER> to go from menu to menu in the editing process. When asked for the "Secondary" this refers to the drive number specified in the devices window for the E/A central menu files. If you ask for a "Reminder", FWB will display the message INSERT UTILITY DISK when you attempt to boot the configured program from a FWB menu. When XB List or UL List configuration is finished (and S(aved) in the case of UL List), press BACK to return to the third window.

THE FINAL CONFIGURATION STEPS: Press BACK several times to return to the second window and then press S(ave) to save the modified SYSCON configuration data file back to disk for later use. Then press BACK, and from the first window press I(nstall) to install the configuration data into the LOAD and UTIL1 files. Follow the prompts. An alternate name for the UTIL1 file is FW and you can use this name is you want. The alternate name used to be RELOAD in earlier versions of FWB, but this name is too long to use with current Horizon Ramdisk Menu software. It is necessary to save the configuration data to BOTH the LOAD and the FW/UTIL1 files, so cycle through the installation process twice. Then press BACK a couple of times to return to the first window and press Q(uit) to return to FWB. If you exit configuration with Q(uit) you will not immediately see your new configurations. It is necessary to reboot FWB from the beginning for the new configurations to appear on screen.

UL LIST SPECIAL CONSIDERATIONS: Immediately after configuring a USER LIST and before pressing BACK to return to the third window it is necessary to S(ave) the configuration to the USER LIST, since this user list data is NOT saved as part of the configuration data file. When you return to the third window your USER LIST data may be lost. You may create as many USER LISTS as you want, each under different names. These lists can be loaded from each other, or they can be loaded from the central menus. DISK UTILITIES from the TI-Writer central menu is a special user list file named DS, and can be configured from the "UL List" option of the third configuration window. If you come across a more recent release of FWB v4.1 you can use your previously configured user lists (files UL, DS, and any of your own user list files) unmodified with the more recent release. You don't have to configure your user lists all over again. I hope it will be possible to use unmodified v4.1 user lists in future versions of FWB (v4.2 etc) as well. Unfortunately the FWB authors state that v4.0 and earlier user lists are not guaranteed to be compatable with v4.1.

NEW FEATURES IN QUICK DIRECTORY:

You can now mark ANY file in QUICK DIRECTORY, invoked by AID from most places in FWB. If the marked file reads PROGRAM, then its name will appear on screen as the default when you load an assembly language PROGRAM file from items 1-3 of the LOADERS menu. If the marked file is D/FB0, then it will show up on screen as the default when you load assembly object code from items 4-7 of the LOADERS menu. The ability to mark files from QUICK DIRECTORY for the LOADERS menu is new to v4.1. Any file may be marked for deletion, and after deletion the sector count and file name list displayed on screen by QUICK DIRECTORY are immediately updated. The ability to delete from QD and immediate updating in both QD and SD (from the editor) are new to v4.1. I consider all the new features described in this paragraph to be very useful.

Other changes in QD include the ability to unmark a workfile name as O(ld) and revert back to the previous workfile name. In v4.0 you could only do this by using SD from the editor. The N(ext) and B(ack) keys are now used to page through the alphabetical list of file names in QD rather than SHIFT/CTRL as in v4.0. This change makes QD consistant with other sections of FWB v4.1 since "N" and "B" are commonly used to move foreward or backward, particularly in configuration.

DM1000 CHANGES:

The FWB authors include their own modifications of DM1000 v3.5 as part of the FWB package. (PLEASE NOTE: DM1000 is fairware, and if you use FWB you should not only send a fairware donation to the FWB authors, you should ALSO send a fairware donation to the Ottawa User Group for the use of DM1000.) V3.5 is the last source code sent directly to the FWB authors by the Ottawa UG and this is why the FWB authors have based their modifications on this rather than a later version. FWB co-author Tony McGovern writes me that he beleives his modified v3.5 will do everything that DM1000 v4.0 will do except line by line scrolling with V(iew). Tony has given DM1000 the squeeze job, and the result is that FWB's modified v3.5 files are smaller than the original v3.5 and much smaller than DM1000 v4.0.

The most important feature of FWB's DM1000 is that it formats disks at 18 sectors per track in DD mode with a Myarc disk controller. Bugs in T(ype), P(rint), and C(opy) have been fixed, and you can now use 3 digit printer entry codes. Horizon Ramdisks at high CRU adresses are fully supported except for initialization during Disk Copy..

OTHER FEATURES NEW TO FWB v4.1:

The formatter may now have 4 disk files open at the same time. From the assembler the object file name is passed back to the object file parts of the LOADERS menu and appears as the on screen default for immediate loading.

The keyboard control of DISK PATCH has been augmented to make it consistant with John Birdwell's DISK UTILITIES. The DISK PATCH title screen tells you that you can use the "original" keys (this means the keys that worked with FWB v4.0, which aren't quite the original DISK0 keys) and an alternate set of key presses that corresponds to the keys used to control DISK UTILITIES (Ctrl/H for Hex display, Ctrl/N for next sextor, Ctrl/W for write to sector, etc). If you are familiar with the keyboard controls of DISK UTILITIES you will have no trouble using FWB's DISK PATCH.

FINAL CONCLUSIONS:

In my opinion everyone who does serious disk based work with the 99/4A should be using FWB. If you don't have it, check your user group's library. User groups, not individuals, may obtain FUNNELWEB v4.1 at no charge by sending a disk and paid return mailer to the Lima User Group, P.O. Box 647, Venedocia OH 45894.

FLEXI LABEL

A TIny Gram

By Ed Machonis

They say the only way to finish a program is to shoot the programmer. I guess the same could be said for the steady stream of label printing programs which seem to come out of this TI-99. I thought I had written all the label printing programs I would ever need, but I seem to keep discovering new needs.

In the past, most of my video tape labeling has consisted of pencil entries on the slip case. A recent visit by my grandchildren resulted in a stack of unmarked video cassettes piled alongside a stack of empty slip cases. They seem to have devised a new game called video roulette.

The only way to restore any semblance of order was to skim through each tape to identify the contents and match it with its slip case. Determined not to repeat this chore after subsequent visits, I decided to do what should have been done in the first place. Label the cassette as well as the case. 20/20 Hindsight!

Mailing labels are an exact fit on the side of the video cassette. Often 6 lines of text are needed for a 6 hour tape with six different programs. Rather than use an existing program, such as Disk Label, I decided to write a more flexible program which could handle Video Cassette labels as well as other types of labels.

Instead of one new program, I wound up with two, each a TIny Gram. FLEXI-LABEL's distinguishing feature is providing the user with the option to print up to 10 lines of text per label. Great for those video cassettes chock full of programs.

When first booted, you are asked to input the number of lines of text to be printed on the label. The font used is expanded compressed which enables an easily readable 28 character line. For labels with more than 7 lines, the font automatically changes to superscript. You are prompted to input the text for each line.

At any time during text entry you can change the number of label lines by entering FUNCTION C (Accent Grave); think FUNCTION C(hange). It can be entered anywhere in a line of text or by itself. The lines you have entered will not be lost, they always default to the next label. FUNCTION 3 can be used to erase unwanted lines. All editing keys are functional. If you want to redo a label, just enter zero for the quantity to be printed.

Text entry is automatically limited to 28 characters. Text can be carried over from label to label without re-entry, handy for those labels requiring only minor changes. Any line can be indented by entering spaces at the beginning of the line. I think you'll find the program as user friendly as they get.

This TIny Gram should answer most of your labeling requirements, whether they be video cassettes, return address, meeting notices publicizing your User Group, "Property Of" labels, or simple mailing labels. Its small size makes it a candidate for your Funnelweb utility disk.


```
1 ! ***** FLEXI LABEL ******
    *      A TIny Gram      *
    *    Copyright 1988     *
    *    By Ed Machonis     *
    **QB-99ers, Bayside NY**

2 OPEN #1:"PIO.LF"

3 DISPLAY AT(8,1)ERASE ALL:"
LINES OF TEXT/LABEL?(MAX 10)
" :: ACCEPT AT(9,26)VALIDATE
(DIGIT):S :: IF S>10 THEN 3

4 E$=CHR$(27):: PRINT #1:E$&
"@"&E$&"G"&E$&"W1"&CHR$(15)&
E$&"C"&CHR$(0)&CHR$(1)&E$&"3
"&CHR$(216/(S+1)):: IF S>7 T
HEN PRINT #1:E$&"SO"

5 DISPLAY AT(1,1)ERASE ALL:"
ENTER ""`"" TO CHANGE #/LINE
S" :: FOR J=1 TO S :: DISPLA
Y AT(J+2,3):"ENTER LINE";J;L
$(J):: ACCEPT AT(J+2+1,1)SIZ
E(-28):L$(J):: IF POS(L$(J),
"`",1)THEN 3

6 NEXT J :: DISPLAY AT(23,1)
:"HOW MANY LABELS?" :: ACCEP
T AT(23,18):Q :: FOR K=1 TO
Q :: FOR L=1 TO S :: PRINT #
1:" ";L$(L);CHR$(10):: NEXT
L :: PRINT #1:CHR$(12):: NEX
T K :: GOTO 5
```

Coded for Epson RX-80

To Be Continued. (YOU HAVE BEEN WARNED!)