SOUTH BAY TI USERS GROUP

BITS & BYTES NEWSLETTER

Mailing Address: P.O. BOX 110037 CAMPBELL, CALIFORNIA 95011-0037

*** JUNE 1989 ***

PRESIDENTS MESSAGE
by Mike Ewell

The demo for the June meeting will be on graphics using the transliterate commands in TI-Writer. If you need a specific question answered, please bring your printer manual and/or a sample of your problem.

The club could use your help with the newsletter. Ask Bill!!
¦+¦+¦+¦+¦+¦+¦+¦ PLEASE HELP WITH THE NEWSLETTER! ¦+¦+¦+¦+¦+¦+¦+¦

If YOU cannot write an article for the newsletter, could you ask a question? You don't even have to sign your name if you don't want to, but the answer to YOUR question may help your fellow club member.

¦+¦+¦+¦+¦+¦+¦+¦ PLEASE HELP WITH THE NEWSLETTER! ¦+¦+¦+¦+¦+¦+¦+¦
The club could use your help with the newsletter. Ask Bill!!

If you have considered buying another brand of computer, but are not sure. Answer one question. Will your new computer do anything that your old computer can't do now? Speed does not count in the question or in the answer. If you had to have a lot of speed, I think you would have already bought your new problem child! Speed can be bought for the 99/4A in the form of ramdisks and internal 32 memory mods for the machine that you already know how to use!! If you don't have your present computer mastered, buying a new one will not help!

¦+¦+¦+¦+¦+¦+¦+¦ PLEASE HELP WITH THE NEWSLETTER! ¦+¦+¦+¦+¦+¦+¦+¦

[]
[] []
[] The June SBTIUG meeting will be held at 7:16 P.M. []
[] []
[] THURSDAY, June 1, 1989 []
[] []
[] The meeting will be held in the Saratoga Public []
[] Library. The library is located at 13650 Saratoga []
[] Avenue. From 280(680), take the Saratoga Avenue exit []
[] SOUTH. The library will be on your left, just past []
[] the Fruitvale intersection. This is about four miles []
[] from the 280 exit. []
[] []
[]

Many thanks to Pat Micetich for her continued support of the club and her labors with the newsletter!

¦+¦+¦+¦+¦+¦+¦+¦ PLEASE HELP WITH THE NEWSLETTER! ¦+¦+¦+¦+¦+¦+¦+¦
The club could use your help with the newsletter. Ask Bill!!

As soon as I get the time, I will convert a common file into both Ti-base and Firstbase for a demo that might better show you which database is the best for your needs.

The club could use your help with the newsletter. Ask Bill!!
¦+¦+¦+¦+¦+¦+¦+¦ PLEASE HELP WITH THE NEWSLETTER! ¦+¦+¦+¦+¦+¦+¦+¦

X NOTICE XX NOTICE XX NOTICE XX NOTICE XX NOTICE XX NOTICE X
X X
X JULY AND AUGUST MEETINGS X
X X
X The July meeting will be held on 7-6-89 which is the X
X first Thursday. X
X X
X The August meeting will be held on 8-3-89 which is the X
X first Thursday. X
X X
X The first Thursday will continue to be the first choice X
X for our meetings, but please read the dates carefully X
X to avoid a wasted trip or missing a meeting. X
X X
X NOTICE XX NOTICE XX NOTICE XX NOTICE XX NOTICE XX NOTICE X

SBTIUG CLUB OFFICERS AND OTHERS

PRESIDENT.....MIKE EWELL...New number>>>415/797-7988
VICE-PRESIDENT.....JOHN WENTE.....408/559-1680
TREASURER.....KEVIN DABERKOW.....408/281-7435
SECRETARY.....NORMAN KNUDSEN.....408/267-6193
LIBRARIAN.....HELMUT FUCHS.....408/263-8339
EDITOR.....BILL SCHULT.....408/446-1182
*ASSISTANT.....PAT MICETICH.....408/227-6212
*ASSISTANT.....??????????????..... 99/4A-9640
SBTIUG BBS.....24-HRS/DAY.....408/258-3679
*Mr. SYSOP.....CHRIS SCHRAM.....408/926-4413
*MRS. SYSOP.....MARTHA SCHRAM.....408/926-4413
VAPORWARE.....TRANS PARENT.....99/4A-9640
PUBLIC RELATIONS.....DON APTE.....408/629-0725
ARCHIVER/NEWSLETTERS..HOY COLE.....SEE AT MEETING

SBTIUG GENERAL MEETING May 4, 1989
by Nora Knudsen

The May meeting of the SBTIUG was held at the Saratoga Public Library and was called to order at 7:21 PM by President Mike Ewell.. There were 16 members and one guest in attendance. One year ago, there were 18 members in attendance.

President Mike Ewell then called for a report from treasurer, Kevin Daberkow, who stated that the current balance was $447.89. There is a phone bill as well as several bills for printing the newslettter yet outstanding. Last years's balance was approximately $500.

Helmut Fuchs has completed a new catalog for the club library.

Our Sysop, Chris Schram, announced that a 6:00 PM there were 742 calls to the BBS.

Has anyone heard that the newsletter needs orininal articles?

Our PR man, Don Apte, gave his usual fine run down on upcoming shows and meets in the Bay Area.

The business portion of the meeting thus completed, we all adjouned for a demo on use of the tranliterate command in TI-WRITER.

TREASURERS REPORT
by Kevin Daberkow

PLEASE look at your mailing label to see if some color has been added. If your membership expiration date has been high-lighted in RED, this is your last issue until you renew. If your membership expiration date is in YELLOW, then you should renew at the next regular club meeting.

>> THE DUES ARE $15 PER YEAR <<

NOTE: Your membership expiration date can be found on the last line of your mailing label.

If any information on your label needs to be changed, please let me know. Call me at (408) 281-7435 or write to me at the following address:

SBTIUG - Treasurer
P.O. Box 110037
Campbell, CA 95011-0037

There were several renewals in the month of May: Charles Nolan, Helmut Fuchs, and Mike Ewell. I would like to thank Charles, Helmut and Mike for their continued SBTIUG support.

The club made the following payments in the month of May: $8,86 to Chris Schram for the BBS phone bill.

The club now shows a balance of $484.03.

Editors Ramblings
by Bill Schult

I would like to thank those that have supported the newsletter in the past with articles. They have been greatly appreciated and I feel that the club and TI community at large has benefited from these contributions. But to continue to produce a quality newsletter the need for future contributions still exist. So if you have an idea for an article, or series do not be bashful, write it up and submit it. The article may be in any format, but if possible I would appreciate it in 60 character columns, but will be accepted in any format.

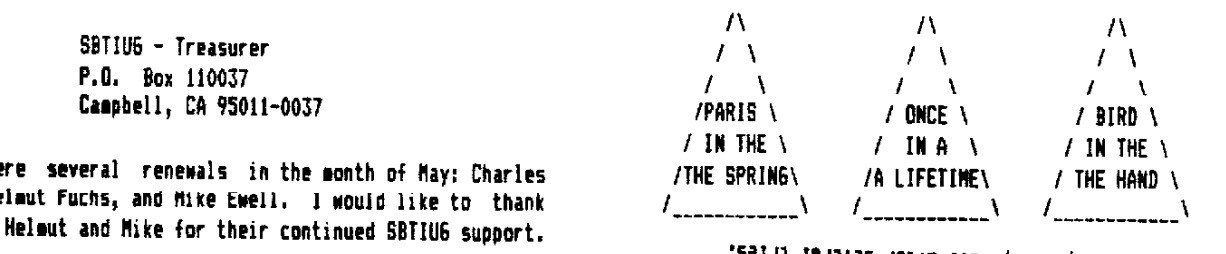
This month we continue with Kevin's series on C language and John Willforths series on disk drives. I am still planning on construction of one of these disk exercizers, just as soon as I find time. Time seems to be one thing none of us has very much of. The disk exercizer seems to me to be something that would be very handy to have around the computer room.

The club is still in need of demonstrations at our meetings. If you have a particular skill or program that you would like to demonstrate, now is your chance to show the rest of us what you have.

Our membership seems to have stabilized at about forty members with about 25 members attending the meetings. I still receive a number of calls for information on our group. We need to find some way to reactivate individuals that have TI-99's stashed away in the closet and get them out to our meetings. The future of the TI community depends on users of the TI. So if you have any suggestions bring it up at the next club meeting.

The Triangle Game
reprinted from TESTS AND TEASERS
compiled by the editorsrs of Readers Digest

If past experience holds true, the odds are 40 to 1 you can't read correctly the simple phrases in the triangles below. Look at each one in turn, and repeat aloud- or write down- what you think it says, before checking the answers below the triangles



Unless you are unusually perceptive (or unimaginative) you repeated or wrote "Paris in the spring" but the triangle reads "Paris in the the spring". Likewise the second triangle has two a's, and the third two the's. An extensive test has shown that only one out of forty persons can read these triangles correctly. Many fail after several tries.

'C'ing is Believing - Part V

by Kevin Daberkow

Last time we looked at how to create a function power(,) to accomplish the task of raising an integer to a positve power. One aspect of C functions may be unfamiliar to programmers who are used to other languages, particularly FORTRAN and PL/I. In C, all function arguments are passed "by value." This means that the called function is given the values of its arguments in temporary variables (actually on a stack) rather than their addresses. This leads to some different properties than are seen with "call by reference" languages like FORTRAN and PL/I, in which the called routine is handed the address of the argument, not its value.

The main distinction is that in C the called function CANNOT alter a variable in the calling function; it can only alter its private, temporary copy.

Call by value is an asset, however, not a liability. It usually leads to more compact programs with fewer extraneous variables, because arguments can be treated as conveniently initialized local variables in the called routine. Below is a version of power(,) which makes use of this fact:

```
/* raise x to n-th power; n>0; version 2 */
power(x,n) int x, n;
{ int p;
  for (p = 1 ; n > 0 ; --n)
    p = p * x;
  return(p);
}
```

The argument n is used as a temporary variable, and is counted down until it becomes zero; there is no longer a need for the variable i. Whatever is done to n inside power has no effect on the argument that power was originally called with.

When necessary, it is possible to arrange for a function to modify a variable in a calling routine. The caller must provide the address of the variable to be set (technically a pointer to the variable), and the called function must declare the argument to be a pointer and reference the actual variable indirectly through it.

POINTERS and ADDRESSES:

Since a pointer contains the address of an object, it is possible to access the object "indirectly" through the pointer. Suppose that x is an integer variable, and that px is a pointer, created in some as yet unspecified way. The unary operator & gives the address of an object, so the statement

```
px = &x;
```

assigns the address of x to the variable px; px is now said to "point to" x. The & operator can be applied only to variables and array elements; constructs like &(x+1) and &3 are illegal.

The unary operator * treats its operand as the address of the ultimate target, and accesses that address to fetch the contents. Thus if y is also an integer variable,

```
y = *px;
```

assigns to y the contents of whatever px points to. So the sequence

```
px = &x;
y = *px;
```

assigns the same value to y as does

```
y = x;
```

It is also necessary to declare the variables that participate in all of this:

```
int x, y;
int *px;
```

The declaration of x and y is what we have been seeing all along. The declaration of the pointer px is new.

```
int *px;
```

is intended as a mnemonic; it says that the combination *px is an int, that is, if px occurs in the context *px, it is equivalent to a variable of type int. In effect, the syntax of the declaration for a variable mimics the syntax of expressions in which the variable might appear. You should also note the implication in the declaration that a pointer is constrained to point to a particular kind of object.

Pointers can occur in expressions. For example, if px points to the integer x, then *px can occur in any context where x could.

```
y = *px + 1
```

sets y to 1 more than x;

```
printf("%d\n", *px);
```

prints the current value of x. Pointer references can also occur on the left side of assignments. If px points to x, then

```
*px = 0;
```

sets x to zero.

Next time I will attempt to tie together the above discussion on pointers with how to write a function whose parameters can be modified.

(Till next time)

'C'ing is Believing - Part V

by Kevin Daberkow

Last time we looked at how to create a function power(,) to accomplish the task of raising an integer to a positve power. One aspect of C functions may be unfamiliar to programmers who are used to other languages, particularly FORTRAN and PL/I. In C, all function arguments are passed "by value." This means that the called function is given the values of its arguments in temporary variables (actually on a stack) rather than their addresses. This leads to some different properties than are seen with "call by reference" languages like FORTRAN and PL/I, in which the called routine is handed the address of the argument, not its value.

The main distinction is that in C the called function CANNOT alter a variable in the calling function; it can only alter its private, temporary copy.

Call by value is an asset, however, not a liability. It usually leads to more compact programs with fewer extraneous variables, because arguments can be treated as conveniently initialized local variables in the called routine. Below is a version of power(,) which makes use of this fact:

```
/* raise x to n-th power; n>0; version 2 */
power(x,n) int x, n;
{ int p;
  for (p = 1 ; n > 0 ; --n)
    p = p * x;
  return(p);
}
```

The argument n is used as a temporary variable, and is counted down until it becomes zero; there is no longer a need for the variable i. Whatever is done to n inside power has no effect on the argument that power was originally called with.

When necessary, it is possible to arrange for a function to modify a variable in a calling routine. The caller must provide the address of the variable to be set (technically a pointer to the variable), and the called function must declare the argument to be a pointer and reference the actual variable indirectly through it.

POINTERS and ADDRESSES:

Since a pointer contains the address of an object, it is possible to access the object "indirectly" through the pointer. Suppose that x is an integer variable, and that px is a pointer, created in some as yet unspecified way. The unary operator & gives the address of an object, so the statement

```
px = &x;
```

assigns the address of x to the variable px; px is now said to "point to" x. The & operator can be applied only to variables and array elements; constructs like &(x+1) and &3 are illegal.

The unary operator * treats its operand as the address of the ultimate target, and accesses that address to fetch the contents. Thus if y is also an integer variable,

```
y = *px;
```

assigns to y the contents of whatever px points to. So the sequence

```
px = &x;
y = *px;
```

assigns the same value to y as does

```
y = x;
```

It is also necessary to declare the variables that participate in all of this:

```
int x, y;
int *px;
```

The declaration of x and y is what we have been seeing all along. The declaration of the pointer px is new.

```
int *px;
```

is intended as a mneamonic; it says that the combination *px is an int, that is, if px occurs in the context *px, it is equivalent to a variable of type int. In effect, the syntax of the declaration for a variable mimics the syntax of expressions in which the variable might appear. You should also note the implication in the declaration that a pointer is constrained to point to a particular kind of object.

Pointers can occur in expressions. For example, if px points to the integer x, then *px can occur in any context where x could.

```
y = *px + 1
```

sets y to 1 more than x;

```
printf("%d\n", *px);
```

prints the current value of x. Pointer references can also occur on the left side of assignments. If px points to x, then

```
*px = 0;
```

sets x to zero.

Next time I will attempt to tie together the above discussion on pointers with how to write a function whose parameters can be modified.

(Till next time)

THE NEW FEATURES OF FUNNELWEB v4.1
by Charles Good Lima Ohio User Group
reprinted from CW/99ers Newsletter

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 word 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
2 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/C6. Items 1-8 in the above DISK UTILITIES user list menu can be altered with CF/C6 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/C6:

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/C6 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/C6 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

THE NEW FEATURES OF FUNNELWEB v4.1
by Charles Good Lima Ohio User Group
reprinted from CW/99ers Newsletter

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 worD 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 DISKO.

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
2 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 DISKO 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/C6. Items 1-8 in the above DISK UTILITIES user list menu can be altered with CF/C6 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/C6:

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/C6 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/C6 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

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/F80, 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 arn'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.