THE GUILFORD 99'ER NEWSLETTER

VOL.5 NO.2 FEBRUARY 1988

Larry Spohn, President
Mack Jones, Secretary/Treasurer
BBS: (919)274-5760(OPUS)
Janice Snider, Vice President
Herman Geschwind, Program Library
ROS (919)-621-2623

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 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Editor)

OUR NEXT MEETING

DATE: February 2, 1988. TIME: 7:30 PM PLACE: Glenwood Recreation Center 2010 S. Chapman Street.

Program for this meeting will be power BASIC using the mini-memory cartridge. There will also be a short program on solving the "programer's nightmare" -- disk not initialized.

ANOTHER REMINDER!! DUES FOR 1988 ARE STILL DUE AND PAYABLE. NO DUES--NO NEWSLETTER ETC!!

MINUTES

GUILFORD 99ERS' USER GROUP MINUTES FOR JANUARY 5, 1988 MEETING

The January meeting was called to order by Vice President Janice Snider at 7:45 PM due to the absence of President Larry Spohn. There were 7 members and one visitor present.

Herman Geschwind presented the disk of the month and four were sold. The meeting was then turned over to Bob Carmany who gave a demo on using the spray de-greaser (Radio Shack) to clean the connections of cards in the PEB, and all other cord connectors that usually give trouble with age. Bob also demoed the TI Diagnostics disk and checked out the club system.

After the program, Bob offered a motion to the club to start sending the newsletter to Hunter Valley User Group and the motion carried. As of the Feb. issue, we will send newsletters to Hunter Valley in Australia.

The meeting was adjourned at 9:00 PM.

Respectfully submitted,
L.F. "Mack" Jones, Sec./Tres.

MODEM TALK

By Herman Geschwind

ROS 621-2623

Let's start off with some statistcs. As I am writing this (Jannuary 9) the total of our download library was 4,442,000 bytes or the equivalent of fifty single sided single density diskettes. Now, since many of the files are compressed, I suppose we can up this estimate to about 65 or seventy diskettes. Quite an impressive collection, wouldn't you agree?

In the download library there is something to suit every taste and every need. In the TIGAMES and TI99 section you will find games galore. TIPERM and TIUTILS are the sections for utilities, languages, tutorial and the like. TIWP is a special section for wordprocessing. Be sure to check NEWIN when you log on to see what is new.

As was mentioned last month, the greatest boon (pun intentional) to TI telecommunications is Barry Boone's ARCHIVER II Vers. 2.3. "Archiving" is the process whereby related files are combined into one download package. Barry's program takes this concept one step further by allowing the archived file to be "compressed". I don' want to go into the details of how this is done, what counts is that after compression a file has been shrunk to only 50 or 60% of its previous size. This means it will take that much less time to download a file. Once the download has been recorded safely to disk, all that is necessary is to reverse the process ("add water and stir"), de-compress and then unpack to have the original, executable files back.

All of the new uploads have been compressed and archived, so having this program is a necessity. In the NEWIN section the file LOAD.TIP is the XB loader and ARC1.TIP is the assembler component of ARCHIVER II. As an exception, these two files have not been packed so that you can get the archiver without having to bother with previous versions.

Here is a quick rundown of what is new:

FUNABRC.TIP and FUNBBRC.TIP are the two disks that make up the latest version of FUNNELWEB. This release is dated 12/11/87 and is supposedly the definitive version of release 4.0.

C4BRC.TIP is the latest release of Clint Pulleys "c" compiler. To get the full set, you might also want to download C99P1BRC, C99P2BRC, C99P3BRC, C99P4BRC, C99P5BRC and CDOCBRC.

MT43ABRC.TIP is the latest version of MassTransfer. Version 4.3 will correctly transmit and receive file headers of archived files. Also the PHONEMAKE program has been reworked if you have an auto-dial modem and need auto-dial support.

MACROABR.TI and MACROBBR.TI is the latest version (6.0) of R.A.Green's stand-alone macro-assembler.

New in the games department is GAME1ABRC.TIG, GAME1BBRC.TIG, GMS12BRC.TI, GMS13ARC.TI and GMS14ARC.TI. These games range all the way from Basic/XB to some very sophisticated assembler versions. Each disk has an XB loader so that all that is required for some enjoyable gaming is the XB cart and joysticks.

LIBRARY.TI is the listing of all the programs in our UG library. If you see something that you would like to have, see me at the meeting our I will upload it.

DUT1.BRC is a demo disk from the Dutch National TI Group that we received via Australia. These E/A programs have an XB menu and loader. Some very interesting graphics programs, particularly the "Lines" variation.

To prevent problems with downloading here are a few tips: (1) use only a quality diskette for the download. Don't waste your time with a beat up "el cheapo" diskette. (2) Format or re-format the diskette and also the work diskettes used for de-compressing and un-packing. The communications programs and the archiver programs use the RAW (read and write) subroutine for writing to disk. This subroutine is fast but pre-supposes a disk without bad sectors. If your formatting shows any bad sectors, don't use it for downloading, compressing or archiving.

Please remember, we also have a message base. Don't feel shy about using it. I just love to read messages and stay in so please drop me a note. Even if you don't have anything earthshaking to say, I welcome any note just to see that you

have visited the board and want to say hello.

ALERT DATA 274-7000

The OPUS board changed systems back in November and still is not quite operational yet. There is a TI message section but the program library for any of the groups, including TI, is still not operable in spite of many promises.

As it is, the ROS board is our "Official" board and there is where you will find the good stuff. Archiving, compressing and uploading takes time and I for one am just as happy to make sure that one board is serviced properly. Dan is doing an excellent job of supporting our group and I think that we are best served by confining our library to the ROS board.

About the only feature that Alert has to offer that is unique is the National TI Echo via FIDO NET. That is a nice way to stay in touch with TIers around the nation. A real good place to place a message if you have items for sale or are looking for something.

ATRAX TRACKS

By Bob Carmany

ATRAX Robustus - a venomous variety of spider indigenous to Australia. It lives in underground burrows with a conical web at the entrance which gives rise to its common name 'The FUNNELWEB spider'.

That brief paragraph is the background of the name 'Funnelweb Farm' which is not only the home to thousands of these strange creatures but also to two of the most creative programmers in the TI world. Will, who was 16 years old at the time, and Tony McGovern created a program some years ago that was meant to replace the TI-Writer module. They named this program 'Funlwriter'. The reason was two-fold. Imagic of Australia realized that they had a "captive audience" after TI pulled out of the home computer market and promptly raised the price on both the TI-Writer and Editor/Assembler cartridges to $200 (Aus). To overcome this rapacious pricing and to guard against the failure of their cartridges, Will and Tony developed Vn 2.0 of 'Funlwriter' as a means to load the TI-Writer disk.

From these humble beginnings, the program was revised and improved until it got to Vn 3.0 which added the Editor/Assembler facility to the program. It progressed on and on until it reached its present form. Version 4.0 will do about anything that you want it to and there are very few programs that it will not load. It has retained the TI-Writer facility and added many improvements and enhancements. So, here we are --- Vn 4.0 dated Dec/22/87!

Let's start by looking at FUNNEWLEB just as you would when the system first 'boots' up. When you put the F'WEB disk in drive #1 and select XB, it will automatically begin. While the title screen is being displayed, a directory program is being loaded. This program, QD, is the same one that is invoked with the (S)how(D)irectory option from the Editor. With the directory loaded, you can catalog any disk by pressing <FCTN-7> at this time. If no key is pressed, the title screen will 'time out' and you will be presented with a menu --- partially filled --- with 18 options. Nine are selectable by number and nine by letter. With the exception of the first three (which are pre-set internal paths) the rest of them can be altered to suit your own tastes quite easily with the built-in CONFIG(ure) program. We will discuss that a little later.

Selecting the first option puts you in the TI-Writer mode. You have your choice of Editor (enhanced from TI-Writer), Formatter (again enhanced), and DM1000 (Vn 3.5). These three choices are best left alone along with option #7 --'USERS LIST'. The rest can be changed and modified to suit your own needs. Option #4 (MODEM) loads program image files starting with a file designated MD. Option #5 (DATA-BASE) loads a program image file designated DB, and Option #6 (UTILITY) loads a program image file designated SP. Although none of the filenames except UTILITY may be easily changed in F'WEB, you can change the names of your favorite programs to the internal paths used in F'WEB and the menu presentations themselves can be altered with a disk sector editor.

Let's choose the EDITOR to begin. It has all of the options that you found in TI-Writer. Some of them have been enhanced and improved significantly. Select 'SD' for (S)how(D)irectory. After you select a disk drive number, you will be presented with a 10-item list 1-9 and 0. You can page backwards and forwards with SHIFT and <CTRL> to show all of the programs on the disk. You can (D)elete files and (V)iew D/V 80 files but two of the most interesting functions are completely new. CHK for check (by pressing the '=' key) will check the programs to see if they are BASIC or XBASIC or E/A program image files. MARK (num) allows you to select a D/V 80 filename for loading by typing in the number. Then, when you select (L)oad(F)ile, the filename is shown as default. The (D)elete function and (V)iew are selected the same way. No more typos when entering filenames!! Incidently, all of the functions appear as on-screen prompts at the bottom of the screen so you don't have to choose them from memory.

The Formatter is similar to the TI-Writer Formatter except that the SD option is available by pressing <FCTN-7>. And,

the printer defaults are permanently changeable when the program is altered with CONFIG.

Option #7, 'USERS LIST', leads to another menu with the possibility of loading another 8 programs. The only restriction is that you cannot use XB programs. But, virtually any Assembly Language program can be loaded from this screen.

When the initial Editor-Formatter menu is presented, pressing the space bar toggles in another menu. This one contains a modified editor (word wrap disabled and assembly tabs fixed), the E/A Assembler (enhanced), LOADERS, c-Compiler (for loading c-99 REL3), DISK-PATCH (enhanced DISK0), UTILITY, and RESET/QUIT.

Option #3, LOADERS, presents a menu of A/L loaders that will load almost any Assembly Language program that you can imagine. There is a TI-Writer environment, GPL loader, program image loader, Load and Run, Low Memory loader, Scriptload (for batch files) and a couple more! It will take care of just about anything you can imagine.

This has been a brief overview of FUNNELWEB. The documentation that come with the program gives an in-depth description and commentary about all of these options and much, much more.

Now that you have your copy of FUNNELWEB, let's start customizing it to suit your system configuration and add some of the programs that you want to use with it. The easiest way to start this project is to copy the following files from the distribution disk to a blank initialized disk. Just copy LOAD, UTIL1, EA, QD, and UL to disk and put it in drive #1.

Next, 'boot' up FUNNELWEB by entering XB as you normally would. From the main menu, select option #2 (E/A) and then LOADERS (option #3). Then, select PROGRAM and enter your drive designation and CONFIG at the prompt to load CONFIG from the designated disk drive. The program will load and you will be faced with four options after following the 'Press Any Key' prompt. They are:

1. CONTINUE NORMALLY
2. REDO USER LISTS
3. REDO CONFIGURATION
4. BACK TO FWB Vn 4.0

You now want to press "1" to start configuring F'WEB to your system. All you have to do is to follow the on-screen instructions to get everything set up. The first choice that you are faced with is to retain or the boot disk tracking. That means that if you choose to have the boot disk tracking active, F'WEB will go to that drive to load the built-in utilities no matter which drive it is (ie. you could load and run F'WEB from DSK3 for example).

The next option is to set up the utility drive default (originally set as DSK2). That means that your text files, for example, will be loaded from drive #2 if you choose to hit <ENTER> and leave things in the default state.

The third option is to change the printer set-up in both the Editor and the Formatter. As the program is set up, they are for a serial printer. Simply enter the printer specifications of your particular printer (ie. PIO and PIO.LF).

The next option is to set up a workfile default. If you choose to leave this blank (and press <ENTER>), the workfile filename will survive a trip through the title screen and remain in F'WEB's mailbox. It is probably best configured this way unless you have a specific filename that you use VERY often.

The fifth option is to change the Option #6 entry in the Editor/Formatter and Editor/Assembler menu screens. The program is set up for each Option #6 choice to be 'UTILITY' with two letter filenames of SP and TI respectively. This is a good time to add SPELLCHECK as a menu option. Simply enter 'SPELLCHECK' and then press <ENTER> to retain the SP filename. Then with a file manager, change the name of UTIL1 on the SPELLCHECK disk to SP. On a DSSD system, you can add the re-named SP file and UTIL2 to your F'WEB disk when you are through and have it available as a menu choice. On SSSD systems, add EA to your SPELLCHECK disk and switch disks when you want to use it.

The next option allows you to change the color defaults that are available by press <CTRL-3> in the Editor. You can retain whichever of them you wish and change the others. A list of color codes is provided. When you are satisfied with your color choices (you have the option of viewing them) press #4 for the next screen.

Next you are given two choices: 'Edit Central Menu UL' and 'Edit XB Load and UL'. The first choice takes you to the editing of the Users List Menu complete with prompts for entering your A/L programs (remember that you can't LOAD XB programs from UL). After you are through, press <FCTN-9> and save the configuration. Answer "N" when the prompt comes up again and and follow the prompts to edit the last 15 choices in the XB load program (the main menu). When you are through, you will be asked if you wish to proceed normally. If you answer "Y(es)", the configuration in each case will be saved and, at the end of the XB configuration, will be saved as either UTIL1 or RELOAD. I would suggest that you read the LOAD docs and the UTILITY docs to decide which name to use to save your re-configured program --there are advantages to each.

If you have followed the prompts correctly, your copy of F'WEB will be configured the way that you want it. All that is left to do is to re-copy the rest of the files that you need (again see the docs) to the disk.

There are some cosmetic changes that you can make to make the presentation of the individual menus more palatable. If you wish, you can use a disk sector editor (I prefer John Birdwell's DISK UTILITIES) to make the changes. It is easy to use and has the advantage of a 'find string' function. In the ASCII mode, search out 'MODEM'. If you look closely, you will see that there are 10 spaces available to enter the name of your favorite MODEM program (I used MASSTRANS for MASSTRANSFER). Right next to it you will find DATA-BASE (I changed that to PR-BASE). Of course, you will have to change your program image

filenames to conform to the built-in loader of MD etc. (modem option) and DB etc.(for the database option).

If you wish to change the BX that Tony used for the file designation in CHK (in the SD function), you can once again change it. I prefer XB because there are very few BASIC program applications left and, besides, they all look the same to F'WEB --- it can't tell the difference between BASIC and XBASIC programs. There will be three places that you need to make this change but it is strictly optional!

That takes you through the CONFIG(ure) process of F'WEB. Next time, we will discuss specific applications starting with each seperate menu. If there are any questions about F'WEB, send them to the UG mailing address and we will try to solve them in this column.

BASIC CORNER

By Tony McGovern

VII. ACCEPT AT and other RAMBLINGS

TI Extended Basic is a very substantial language. The XB cartridge contains 12K of ROM and 3 and a bit (the 4th one isn't full) GROMs at 6K apiece. This is on top of the 8K of console ROM and whatever parts of the 3 console GROMS are still used in XB. The tragedy of the TI-99 is that GROMS and GPL were ever invented. I guess it was TI's way of trying to keep the software market sewn up. The end result as we all know is that they shot themselves in both feet with uncanny accuracy. Instead the TMS9900 CRU addressing to bank switch plain ordinary ROMs or even just used GROMs only as sources of code to load into RAM, they could have had a machine that did justice to its CPU, a real home minicomputer ..... that's all past history now.

I have been pondering on what TI should have done way back when the 99/4 was first designed, that could have been easily done at the time (or even when it was updated to the 99/4a). My conclusion is that the machine should have been given 4K of fast 16-bit CPU RAM instead of a measly 256 bytes. There would have been plenty of room with a little rearrangement and/or better decoding of memory-mapped devices (VDP, sound, speech, GROMs). This would have meant that Basic and XB system areas, sprite tables, full screen buffers, string buffers, value stack, and so on could have been in fast RAM, and even console Basic could have had full scope for character and sprite definitions (as in TI-LOGO for instance). Their cartridges could then have easily been a lot better, and let's face it, many of the earlier ones were pretty hopeless, and the later ones are all limited by lack of honest CPU RAM. Really good programs have only just started to appear (TENNIS for example), a year after TI laid the 99/4a to moulder in its grave. TI would then have never been dragged into that marketing war to the death (TI's that was) with that vastly inferior machine, the VIC-20. I have a suspicion that the 256 bytes happened because part of TI management wanted to protect their existing evaluation board and smaller minicomputer business.

Proposed specs for the CorComp 99/64 are floating around the User Groups now. Perhaps this is the 99/8 rising from the ashes. It reads well, but I have a few reservations. The VDP sounds just like the 99/4 VDP rather than the more advanced model that TI is working on. And only 64K RAM as standard in a new machine in this day and age ? They may be thinking of it as a way to sell high profit sole-source memory cards, and if so they haven't learned anything from the TI-99/4a's demise at the hands of Commodore. I hope that 99/4 compatibility does not mean that GPL and GROMs form any part of the normal operation of the new machine. The biggest reservation I have concerns the Enhanced, Extended Basic. This looks very powerful but has one disastrous omission, as it does not support honest procedures, sub-programs with local variables as in XB. That's not enhanced, that's just caught the Microsoft / Apple / IBM PC Basic disease (Commodore Basic is beneath contempt). Genuine procedures are a major requirement for a good Basic. TI Extended Basic (showing its engineering/instrument company heritage) has had this feature for years now. I suspect TI's programmers who obviously put enormous effort into doing the user subprograms properly were disappointed that so little good use was made of their finest feature in magazine articles and books. What was needed in XB was for user defined functions to be upgraded to the same level of performance as subprograms, and editing improved.

If CorComp has really given up this feature (enhanced ??) then I may very well pass their machine by, despite its TI-99 compatibility, when the time comes to upgrade. Other Microsoftish clutter seems to have crept in too. As it is I am waiting until the Intel 8088 has faded to an unpleasant memory. The immediate improvement really needed in XB sub-programs is a means of examining variable values in any sub-program when program execution is halted by BREAK or errors. TI should have done it in XB by retaining the EDIT command of console Basic, allowing it to access user subprograms by name. Anyone listening out

there? If so add single command array operations, full syntax checking on entry, 80 column display capability with formatting power to match, bit-map screen functions, fast program execution and anything else will then be gravy. Then TI-99/4a owners will be most pleased to join in. The bad news is that TI is starting to cut back on support for the 9900 family despite its excellent qualities, and so it is becoming less attractive for new designs.

Enough ramblings and back to the tutorials ! What then is the most powerful feature in XB after SUB and CALL? A good candidate is the file system, but as this is already built into the console I will stick with commands specific to XB. The prime candidate is ACCEPT AT and its qualifying clauses. This was emphasized by the recent appearance (mid-84) in a computer magazine of a long article on machine code for adding this function to IBM PC Basic (which doesn't have sub-programs either). ACCEPT AT is very useful and powerful, but has some undocumented features as well as some subtle and treacherous bugs, and is well worth talking about in this series.

The simplest level of ACCEPT AT combines the INPUT routine with its access to editing features, with cursor positioning on the display screen by the AT clause. So far this is just the input version of DISPLAY AT. The difference from INPUT is that there is no provision for prompt strings, but a DISPLAY AT soon fixes that. It also accepts input to a single variable only, and not to a whole variable list. As ACCEPT AT and DISPLAY AT do not scroll the screen, their repeated use can give a much better effect than INPUT when graphics elegance is important. Construct your own examples here or work the XB manual examples. Remember that the cursor is in XB color group 0 if you are trying to dress up the graphics.

BEEP allows an audible prompt with only one program byte (we'll talk about program length later on if it keeps going long enough). Of course constant repetition of beeps can get a little wearing. The ERASE ALL clause provides an alternative to CALL CLEAR for clearing the screen. As compared with CALL CLEAR, ERASE ALL is slower to execute, (it seems to be line at a time) but takes less program space. Its effect is slightly different also. This little program which uses ERASE ALL with DISPLAY will make both speed and screen effects easy to see.

```
100 CALL CLEAR :: CALL COLOR(0,3,3)
110 FOR I=1 TO 100 :: CALL CLEAR :: NEXT I
120 FOR I=1 TO 100 :: DISPLAY ERASE ALL :: NEXT I
130 CALL SCREEN(11):: FOR I=1 TO 1000 :: NEXT I
```

That's the simple pieces of ACCEPT AT -- now it starts to get interesting. VALIDATE allows the programmer to decide what characters are acceptable in a response. The computer honks (that's the word in TI-FORTH) at unacceptable inputs. Three predefined types are available. UALPHA accepts only upper-case alphabetic characters -- very useful for filenames and suchlike. This is not quite the same as depressing the alpha-lock key as it it only accepts letters, and so is incompatible with input to a numeric variable. If you are in the habit of verifying wet paint signs by touch, try that for a change. The DIGIT type does just what its name implies, and NUMERIC allows the input of any floating point number as well as plain positive integers. As with INPUT, all numbers are acceptable to a string variable, but numeric variables are fussier.

Now what if these predefined types aren't right for what you want ? Suppose only digits 1 to 4 are acceptable, as in a menu choice of 4 items labelled 1 to 4. In console Basic extra lines of code would be needed to check the input, but ACCEPT AT handles this with the clause VALIDATE("1234") or VALIDATE (I_LIKE_IT$) where the string variable has previously been set to "1234". To put it more formally, only the characters in the string argument of VALIDATE can be entered at the keyboard to be ACCEPTed.

The SIZE clause allows ACCEPT AT to be used with almost no interference to screen displays. It blanks out the specified number of characters, providing an input window of finite length, and if the length specified is negative, the characrers already in the window are not erased, and form an immediate input for ACCEPTance. This is very handy for making default choices obvious to the user. Let's enter a little program to get at the essentials.

```
100 CALL CLEAR :: DISPLAY AT(12,1):RPT$("_",28)
200 ACCEPT AT(12,2)SIZE(3):A$
300 DISPLAY AT(15,2):A$;LEN(A$)
400 CALL KEY(0,K,S):: IF S>0 THEN 100 ELSE 400
```

You most likely have the Alpha-lock depressed. If so let it off, and RUN our little program. Just press ENTER the first time round, next time hit <space> first, and finally <space> first before hitting another key. This shows that <space>s after the last honest character entered are ignored. Try some VALIDATEs here too, if you wish. Now with the program as given, alter SIZE(3) to SIZE(-3). It now ACCEPTs whatever is in the was or is placed in that 3 character input window.

Now that's all very simple, but it brings us to the edge of the undocumented wilderness. Alter the CALL KEY(0,K,S) in the last line to CALL KEY(3,K,S) and RUN the program again, this time entering letters. Observe what happens the second time around. This answers the question of what keyboard mapping ACCEPT AT uses -- like CALL KEY(0,K,S) it uses the last one, whatever that was. Try split keyboard units in the last line. At the machine code level, a particular byte in the CPU scratchpad RAM has to be set to the key unit before calling the SCAN routine. I interpret the behaviour as showing that in the XB modules of my experience that ACCEPT AT does not alter this byte. The XB manual however does not document this behaviour at all. If XB weren't a dead language that would be a caution signal. It does need to be watched in your programs, if your last CALL KEY wasn't the key unit you want for ACCEPT AT. On the positive side you can control ACCEPT AT with a prior dummy CALL KEY to ease input for the user. An example is when a program requests input of a filename, setting the key unit to 3 makes letters come out as upper-case while still allowing other characters. Brian Rutherford first brought the anomalous behavior to my attention.

Now that's not too bad, but there is worse to come. Insert a VALIDATE("123") clause in the ACCEPT AT and RUN the program. No problems there with SIZE(3), but SIZE(-3) is trickier. You can't enter invalid characters from the keyboard but unaltered "_"'s slip through. The VALIDATE appears to be exercised as characters are entered from the keyboard, and not as the edit buffer contents are transferred into the target variable. The decision to ignore trailing blanks in the input window is taken then however. Presumably a negative SIZE pre-loads the edit buffer with the screen window contents without doing a VALIDATE check. Ultimately this is not a real problem since the programmer can control what is on the screen before ACCEPT AT is invoked. Once again, the XB manual does not bind ACCEPT AT to work this way.

This behaviour does leave a weak spot in ACCEPT AT which can only be considered as a bug, but not an intractable one. Suppose you have a menu choice of items, say 1-4 by number, with default 1 pre-loaded in the SIZE(-1) window, and a VALIDATE("1234") clause to ensure proper entry for a numeric variable . What can possibly go wrong ? An evil-minded program tester would immediately delete the default using FCTN-1. An attempt to enter the blank will then cause the screen to scroll with a WARNING message. This is not a fatal error, but might as well be if your background is a carefully composed graphics screen. The workaround for this problem is not difficult, but the best one also resolves an even worse bug, so I will leave it for a little while. I do consider suppression of error trapping or warning messages by global ON ERROR or ON WARNING to be poor programming practice. The best safety net is one that is never used, only tested.

Now go back to the original sample program and change every every A$ to an array element A$(2). Default dimensioning will do. Nothing changes. Next alter your A$(2) in the ACCEPT AT to A$(1+1). Now it works only if there is also a VALIDATE clause, but the SIZE window is disabled and input can even spill over into the next line. No, it's not useful as a multiline ACCEPT ! The solutions to this and the previous problem are the same --- always ACCEPT into a temporary simple string variable, and then process the return, and do not ACCEPT a numeric directly or ACCEPT into an array element with computed index. Both of these problems were turned up by my testing crew during the writing of TEX-BOUNCE, and served as a reminder that program testing should never be left to the author of a program. The same holds true for writers of languages!

Might as well keep on going with the entomology lesson. The sub-program CALL ERR fails to clear errors when the DSR routine cannot find the external device, as in attempts to access an empty disk drive. The work-around this problem is to have a second bash at CALL ERR after further trying for a file on the device which failed to OPEN. The OPEN cannot be CLOSEd without crashing the program or invoking this extra step to flush out the Peripheral Access Block. The Feb/85 Newsdigest carries a letter giving that essential extra step. Also Ross Mudie called up earlier with a similar approach. I had put it aside as something to be dealt with from Assembly language, which is the way I will do it in the next update of COLIST.

The instruction ON BREAK NEXT is useful, particularly in games, for disabling the FCTN-4 (BREAK) key action. However a CALL SOUND with duration greater than 33 over-rides that. Just why is not so far obvious to this outside observer.

REFUND ! ! !

Here is just a quick note to those of you who are dissatisfied with THE SMART PROGRAMMER and its complete lack of anything resembling a regular publication schedule. YOU CAN GET A REFUND for the unfilled portion of your subscription. The easiest way to do it is to write to Bytemaster Computer Services and request the termination of your subscription and the refund of your money. The address is - Bytemaster Computer Services, 171 Mustang St., Sulphur, Louisiana 70663-6724. Another address that might be of value is - Better Business Bureau of Southwest Louisiana, P.O. Box 1681, Lake Charles, Louisiana 70602-1681. The money that you get could be much more constructively invested in one of the other, more timely, publications for the TI or a membership in one of the more active Users' Groups. After all, most of them offer a monthly newsletter,

produced on [illegible] and usually with a better range of topics than was offered in SP.

[illegible] UG newsletters, here are a few that are well worth the subscription price for a non-attending member: 1) West Penn 99ers, RR #1 Box 73A, Jeanette PA 15644 (technical info), 2) K-Town 99/4A Users Group, 3506 Garden Dr., Knoxville, TN 37918, 3) Hunter Valley 99ers UG, 6 Arcot Close, Taro, New South Wales, Australia 2322, 4) L.A. 99ers, P.O. Box 3547, Gardena, CA 90247-7247, 5) San Diego Computer Society, P.O. Box 83821, San Diego, CA 92138, and 6) Chicago Users' Group, P.O. Box 578341, Chicago, IL 60657.

For a list in your own area, see the May issue of MICROpendium. There was a lengthy list published in that issue and updates and address changes are published on a monthly basis. COMPUTER SHOPPER also carries a list of Users' Groups on a monthly basis and, between the these two sources, you should be able to find one to your liking.

Or, better yet, take that refund and send it to one of the "fairware" software authors out there. Over the past couple of years, there have been some outstanding software packages produced for the TI.

PEEKS AND POKES

Christmas and New Years' eve have passed on, and now it's time to get down to concentrating on our activities coming up. First off let me say I will try to serve you the very best I can as your new Secretary. I intend to try and have the newsletters from the various clubs, that we swap newsletters with, available for you to check out each month. I feel that they contain a world of information for us all.

The newsletter last month was the first for George Von Seth and I think he did a bang-up job for us. Thanks for taking it over George. Mike Garrett deserves a vote of thanks for the good job he has done over the years with the Sec./Tres. job. With all the TI goodies he bought and all the vacations he took, the books still balance!! By the way, Mike got us some back monies that were mistakenly charged us when the bank changed our status to a business 7 or 8 months back. We got enough for about 2 memberships out of them thanks to Mike. If you want to know how to jump Bank folks, just watch Mike in action!

We have received quite a few letters lately from TI owners who have seen our Club advertised in Computer Shopper. One lady wanted to order a printer and Speech for her TI from us!! I guess she thought we were a business. I sent her a letter telling about the Club, but she lives in Lexington, so that may be just a little far for her. You would be surprised at some of the requests you get from users who write us.

Before Christmas, I downloaded a program that was on ROS about Snoopy's Christmas present to Woodstock. It was by Ray Kazmer and a darn good one I must say. I sent Ray a couple bucks and in a few days before Christmas, I got a disk of games and utilities from Ray who wrote that he always sends a free disk to the first sender of monies for his software. There was also a page or two that told how he makes drawings on the screen using wax paper and TI Artist. He asks that if we print this, that we send him a newsletter showing it. I surely don't think that's too much to ask and I will give the info to George and see if he has room for it. If not this month, then maybe next. Ray has done some real good pictures and his favorite people are Garfield and Odie, so guess what the most pictures are about!?

Lastly, I want to remind you that you only have one month grace period for paying dues, and the January Newsletter will be the last you will receive if your dues are not in by the end of this month. I hope that each one of you will place a high value on your membership, because when I read letters we get from users in towns where they have no club and want to join one long distance, it makes me feel really fortunate to be able to belong to our group. Stay with us because there are new programmers out there just waiting to give us something entirely new. New hardware items are just around the corner, and we are now able to reach most of the states via TI ECHO on the OPUS BBS. (274-7000). You wouldn't believe the downloads possible on ROS (621-2623). Both Herman, and recently, Walter Tietjen from Raleigh have uploaded many good programs for us to download. I hope you can join us for another year.

Any time I can serve you, you have only to call me at home (288-4280) or see me at the meeting. I will help you in any way I can. Thanks again for voting me your Secretary/Treasurer, and have a very good New Year. (Submitted by "Mack" Jones)

FORTH TIPS

By Bob Carmany

One of the misconceptions about TI-Forth is that it is difficult to have on-screen documentat programs. That's true only if you try to do things the hard way. Messing around with GOTOXY and text think that maybe documenting your programs isn't the easiest thing in the world to do. In fact, u of like trying to swat a fly with a baseball bat --- it can be done but it's hardly elegant!

Luckily, there is a much better way to do it! TI-Forth has, as part of the resident vocabulary, t MESSAGE can be used in a loop to print anything you want from any screen without having to deal with t There is only one catch --- everything is relative to Screen #4, Line #0. Complicated? Not really!

First of all, you should have the message screens (#4 and #5) installed on every TI-Forth disk that y those positions. That way, if you encounter an error, you will get the appropriate message rather than "go

Let's assume that you have finally typed in and debugged that "dynamite" application that you have been working on. Let's say that you started at Screen #10 (on a SCREENS disk) and ran through Screen #15. Now, you are ready to add the documentation on Screen #16. How do you go about it? Quite simply, actually! Just start typing in the documentation on the very first line of Screen #16 --- no quotation marks or any other preparation. The only restriction is to make sure that each line doesn't exceed 40 characters so that it will fit on the screen when it is displayed. Once we are done, the only thing left is to calculate the values for our MESSAGE loop and design a word to access the documentation.

The calculation for the beginning of the loop is really easy to do. Since everything is relative to Line #0 of Screen #4, and there are sixteen lines per screen (0 - 15), all we have to do is to subtract 4 from the screen number (16) and multiply by 16. We get 192 as the answer. That is the value for Line #0 on Screen #16. If our documentation stretches for some 10 lines, the other value will be 202 (192 + 10).

Now, we have to go back and design a word to access the documentation. How about HELP? That seems to be the universal word to use.

```
: HELP CLS 0 0 GOTOXY 202 192 DO I MESSAGE CR LOOP ;
```

There we are! Just squeeze this word into Screen #15 somewhere near the end. But let's dissect HELP before we go on. The "CLS 0 0 GOTOXY" starts by clearing the screen and printing our documentation at row 0, column 0 but the value could be easily changed to suit your own preferrence. "202 192 DO I MESSAGE CR LOOP ;" is the remainder of the definition. Since Forth uses a "last in, first out" method of storing values on the stack, the higher of the two line numbers is entered first. All this definition does is to start printing our documentation at coordinates 0,0 and loops through lines 192 to 202 (our docs) with a carriage return at the end of each line. No GOTOXY and text strings for each line and it is much more elegant to use.

The only problem is trying to calculate the values for the starting and ending lines of your documentation. If your math is as rusty as mine is at times, the following program should help you out. Just follow the prompts as they appear on the screen and write down the values to use in your MESSAGE loops. That's all there is to it. Who says that writing on-screen documentation for your Forth programs is difficult?

```
100 DISPLAY AT(2,3)ERASE ALL:"MESSAGE LOOP CALCULATOR"

110 DISPLAY AT(6,4)BEEP:"Screen number " :: ACCEPT AT(6,18)SIZE(3)VALIDATE(DIGIT ):SCR :: DISPLAY AT(8,4)BEEP:"Line number "

120 ACCEPT AT(8,17)SIZE(2)VALIDATE(DIGIT):LNUM

130 LN=((SCR-4))+LNUM

140 DISPLAY AT(12,2)BEEP:"Screen# " :: DISPLAY AT(12,10):SCR :: DISPLAY AT(12,14 ):"and Line # " :: DISPLAY AT(12,25):LNUM

150 DISPLAY AT(13,2):"is Line #" :: DISPLAY AT(13,11):LN :: DISPLAY AT(13,14):" relative to" :: DISPLAY AT(14,2):"Screen #4 and Line 0"

160 DISPLAY AT(23,1):"Another (Y/N)?" :: ACCEPT AT(23,16):CHOICE$ :: IF CHOICE$= "Y" OR CHOICE$="y" THEN 100 ELSE 170

170 FOR DELAY=1 TO 1000 :: NEXT DELAY :: END
```