

Vol.8 No.10 December 1990

NH

NEW HORIZONS



NEWS LETTER

NORTHWEST ONIX COMPUTER CLUB FOR THE TEXAS INSTRUMENTS 99/4A  
AND THE NYARC GENEVE 9640 PERSONAL AND HOME COMPUTER

This newsletter is published by New Horizons TI-99/4A Home Computer User's Group. Material may be reproduced without permission provided that the Author and the source are Acknowledged. For more information consult one of the following officers: Yearly Dues \$15.00 per Family or Individual.....  
THIS MONTHS MEETING DEC.08, 1990 SATURDAY AT UNITY CHURCH 12:30 PM.  
Behind Wendy's off Secor Road on Executive Dr.

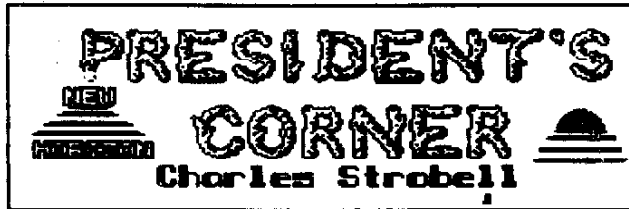
TI-CON BBS.....	1-419-385-7484
PRESIDENT... CHARLES STROBELL	1-419-829-3527
ADDRESS 1225 SCHMADERGER RD, HOLLAND	
ONIX 43528	
VICE-PRES... BUD MILLS	1-419-385-5946
SECRETARY... MARILYN SCHAFSTALL	1-419-882-6870
TREASURER... EARL W. HOFFSIS	1-419-475-0461
SOFTWARE I.R. SCHMIDT'S GENEVE	1-419-475-3871
NEWSLETTER LIB. BIAN MALLORY	1-419-882-6769
EDITOR... ROGER FEINHAUER	1-517-263-6144
COMPILER... JUDY FEINHAUER	.. .. .
TEK SUPPORT. DAN TURNER	1-419-474-6135

MERRY CHRISTMAS FROM ALL OF US TO YOUR'S.



N.W. ONIX 99ERS USER GROUP  
271ST CHURCH UNITY  
3535 EXECUTIVE PARKWAY  
TOLEDO ONIX 43606

Dallas TI Home Computer GP  
PO Box 29863  
Dallas, TX 75229



# Vote

Merry Christmas to one and all, and I hope each and every one all the best for for the rest of this year and the next.

Some thoughts about last months meeting. How many tried TI-Writer? I hope some of you have. If so please bring what you have been working on to the meeting. Also if you are having any problems with it bring these to the meeting also. The funny thing about the software is if you don't work with it you only learn about one-tenth as much as if you used it. So bring it with you.

Roger Feinauer will be doing a demo with Archiver 303 which will be interesting. Also we will have more on TI-Write, so you may want to bring pencil and paper so you can take notes. This will help you to remember what went on in the demos.

The club disks will be on sell as usual. So help the club by the club disks. If we don't sell many. This may be a very lean year indeed. MICROpendium will also be on sell as usual.

We will also have our 50/50 drawing, and the drawing form the drawing program.

If you know anyone who is interested in bying a computer but doesn't know what he wants bring him to the meeting. He may just find that the TI is just what he is looking for. "SEE YA!

During the Dec. meeting we will be taking Nominations and voting for club officers.

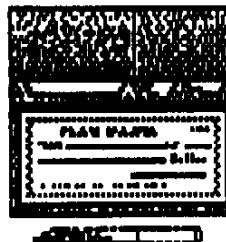
The following are nominees that have accepted:

- FOR V.P. Bud Mills
- FOR TREAS. Earl Hoffsis
- FOR SECRETARY Marilyn Scharstall

The officer of President is at this time without a nominee.

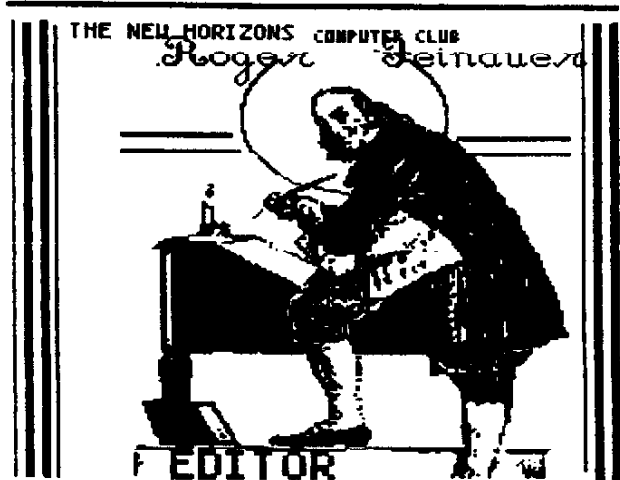
Everyone should attended this meeting and partake in this election. All nominees must be paid up members for the year of 1991

Jo Symington  
Nomination Comm.  
Chairperson.  
phone 1-419-474-4128



Club dues are due for the 1991 year. Cost is \$15.00 per member or family. What do you get for this sum you may ask?

Well you get to preview the most recent software that IS still coming out for the TI99/4A. If you have trouble with something that doesn't work right or you can't seem to figure it out maybe we can help.

Page  
03

Well this marks my last issue as your Newsletter Editor. And through it all I have had a wonderful time. When I started all I new about writing on TI-WRITER was how to load and save files with the text editor, now I know how to use the text editor, the Formatter, and how to use the Control U functions to the max. And along with this also a little bit of programming in Basic and Extended Basic, mostly to do things faster or better with my text files. \

I know a lot of times I sounded a lot of gloom and doom. But, most of the time I was just showing a soft spot for our computer. \

Right now I find myself really excited, because I have been looking at other country and am finding that there is a lot still going, on as far as new software is concerned. The only problem is the information isn't getting spread like it use to. You people if want to see some really great stuff see if you can get someone to show off there 80 column card for the 99/4A. You say there isn't any software to run. Guess again, Funk/Webb will now run in 80 column mode. So will Telco, And now there is a new program

call YAPP from Asgard Software. That uses 182K. of the 9938 chip. This is even more then the Geneve has on board. It has most of the functions the Geneves Myarc program + more. It has OSR's for most of the input devices such as both Myarc Menitech nice, joysticks ect. The program will also load Ti-Artist Fonts for its text in the picture also will load the boarders set to put in boarders. the program at this time won't load Artist Instances as yet but you can fool the software into load them. Load the E/A editor and load a Instance file. On the first line there are two numbers such as '10,14' well add a third number such as '10,14,10', then press function 8, to insert a line. At the left most column put on 'A' there. Then save the file in DU 80 such as DSKn.NAME\_F WHAT YOU HAVE CREATED IS A SINGLE CHARACTER FONT FILE. Which will load your graphic clip art. Also Mike Dodd's Identifile program will run in 80 column mode, Macflix, and the Boot program to mention a few. At this time I would like to thank Don Block for the EAS loader for Adventure nodual now I won't ever need this cart anymore.



Let's turn the page,  
and finish this race.

Thoughts on TI-Writer  
by:  
Roger Feinauer

Good day to everyone and hope every one is doing fine in every way. Well last month we started to study the many uses of TI-WRITER and hoped what you seen from Earl's demo last month was enough to get you started on your writing endeavors.

One important thing to remember is that TI-Writer. Has three modes, when the editor is loaded. Most Versions load in what is called word wrap mode. This can easily be told by looking at what shape the curser. If it is a solid box then you can assume that it is in Word Wrap. What does this mean you might ask, well, for one thing in this mode when you get to the end of a row and in the middle of a word the computer will automotily carry the whole word to the next line. In this mode you will also find that if you insert a letter or word, by using Function 2 you find that everything after the curser drops down one line. This inablaes you to insert one letter or a whole phrase. Then if you want to reconnect the word or phrase back that which was dropped down one line. Just press Control 2 and what seems like magic everything seems to be sowed back up like it was, neat ha!. One last item in Word Wrap mode never hit enter untill you have finished a parograph. This is what the program uses to lock in a group of words for word wrap. When you hit enter in this mode you will see a

character that looks like 2 tiny characters packed together that looks like a little cr. This stands for carriage return. This is were we can look back to the insert and as the manual calls it reformat, or putting it back together again. Well if you are not carefull in this mode you could insert find a jumbled mess. One case is if you have different length of lines of text. Such as lets say you are writing an article about a basic program and in the text you wise to make the program look like it does in Basic or Extended Basic. Sense they both have a 28 column editor you would want the text to represent this to be only 28 columns and say the rest of your text would be set 32 columns. How is this done, well I will get into this in a little but. And say you just got through proof reading and discovered you nissed spelled something above. But you didn't separate this small area with the little cr's. Well you went up to correct the miss spelled word. The insert worked ok, but when you did the reformat, and to your suprise all the 28 column text was turned into a 32 column mess. Because you didn't end the last part of the 32 column text with the little cr's, SOMTHING TO REMEMBER. Well now to press enter to end this parograph for the next.

Well, maybe I should mention a little bit about Tabs. They let you set the column width of rows of text. Also set the indents for the begining of parographs, and in some cases even set a bell for end of margins. To access these, and as a matter a fact all the editor functions you must go to the



command menu. To do this from any where in the editor simply press Function 9. You will be greeted with a line at the top of the page with some options. To do this you will notice that all of the options have both some upper and lower case text. To make any of the active you can either type in the whole name or just the upper case characters for each command. The only important thing to remember is you have to be at the command menu to make them work.

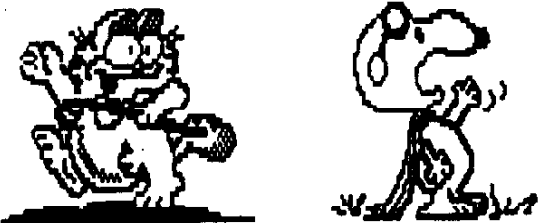
To set the Tabs press function 9, and ether type Edit Tabs, or just simple ET, ether will work. The screen will change at the top of the screen only. And a dotted ruler will appear. With numbers 0 thru 7 will divide it up. To set the Tabs put a L were you want to set the left margin and a R were you want to right margin set. Also for paragraph indents place an I right of were you placed L the amount of spaces you want the editor to automatically set at the begining of each paragraph. So for this article I set the Tabs L-0 I-2 R-28 and because i'm using the Funl/Webb disk that was sold at the last meeting it has a bell option which I set B=25 just to let me know I was at the end of a line.

The second mode for the editor is the Line Mode or Program mode. To change to this mode simply press Control 0. Again you can tell you are in this mode by the shape of the cursor. If it is a hollow out line character then you are in line mode. In this mode word wrap is turned off so when you get to the end of the line, you can keep on typing but the last letter of the line just gets replaced by what ever you typed. To get

to the next row you must press enter. This mode is most use for text that must be save without control codes. Such as the Editor Assembler, Fortran, small c, Font Writer II or the like. Their is more it can be used for such as merging portions of text form other file or saving portions of a file to be used latter.

the last mode is called Control U. And again the Curser changes again to an under line character. To change it back simple press control U again, and it will return back again. This mode is used to imbeded printer contol characters in the text.

Three last commands are Load File, Save File, and Print File. And again these can be accessed by ether typing in there full name or just those parts capital letter. Well thats all the room I have so till the meeting. See Ya!



**These crazy dogs, i've got one at home thats just as silly.**

Tip on Page Pro on the Geneve, because of the speed difference you may find that when doing a directory the system doesn't respond to the space command to read more files. Well the problem is you released ether the control key or the "C" too soon. Just hold the key a bitt longer and the command will work as it should.

## QB MONITOR ~ QB-99'er NEWSLETTER

Reprinted From HOUSTON User Group (HUG) Newsletter

### PROGRAMS, FILES, AND LOADING

By Richard Lupkin

The following information is a merging of various articles from: Topics, LA 99ers; "Gil" Gilmore in the Aug 86 Shoals Tidings, via SMAUG/99, May 88; San Diego TI SIG, Oct 87, Woody and Tigerclub TIPS, plus a reprint of Jerry Bentzinger of the Oaaha TI UG, May 1986; Joe Makeel, Northcoast 99ers; Rich Klein of the Chicago Times; and some other random sources and old notes.

One of the larger problems for the novice TI'er, or for most any Computer User, is HOW TO GET THE PROGRAM INTO THE MACHINE AND STARTED. The most common source of programs for the 99/4A these days, and the ones where the trouble starts, are the Disk programs. These are a problem because virtually All FORMATS of File Storage are used and found on disks, and because of the wide range of Programs being created and distributed on Disks. SO.....as a cookbook guide to Programs and the Operation Thereof, I will contribute....

The original question of "HOW" is directly related to the question, "What have I got on this disk?" The general answer for the 99/4A, and for most computer disks, is that the disk is inhabited by "Files"--which fall into three categories, based on CONTENT:

1. Text Files--those devoted to 'human' languages.
2. Program Files--those devoted to 'computer' languages and which 'operate' the computer.
3. Data Files--those with multiple occurrences of similar groups or types of information, and which are accessed or used by Other Programs.

Text Files can be anything which is produced or read / printed by a Text Editor Program--letters, memos, BBS messages or downloads to be read, etc.

Program Files can be in various FORMATS, depending on the type of computer, and can be in various computer languages--BASIC, PASCAL, FORTH, FORTRAN, LOGO, etc, or the Machine Language of that particular computer, either as "Source" code or the resultant "Object" code file.

Data Files are those which contain INFORMATION, which is (random) accessed by the Programs). It may text, program commands, machine language numeric values, stored number values, pictorial or graphic data files, etc., and may be

stored in almost ANY FORMAT or FILE TYPE, depending on the Creating Program.

A. FORMATS of files on disks are dependent on the computer; the 99/4A uses FIVE storage formats or "shapings" for Files: Display, Variable (length); Display, Fixed length; Internal, Variable length; Internal, Fixed length; and Program. FORMATS ARE NOT SPECIFIC INDICATORS OF PURPOSE OR METHOD OF USE for a File. Some "Display" files are NOT for display, but rather are Programs which "Run" the computer. Some "Program" files are actually Data or Text Files. The USE of a File may have to be deduced from context or Name of the File, or from other indicators such as size, related Program files, etc.

B. FIRST STEP with any DISK or Disk-based PROGRAM is, CATALOG IT. Then you can see what Files you have and maybe pick up a few clues to the Load and Run procedures from the names and types of files on the disk. Most D/V-80 files are TEXT files and sometimes contain the documentation or instructions, in which case they may be named "XYZ/DOC" or "READ-ME", which is exactly what you should do, or better yet, print them off with TI-Writer or a "File Reader".

DISKS CATALOGs list out as follows:

Filename	Size	Type	Rec	P
----------	------	------	-----	---

Filename==this indicates the NAME of the File; it will be referred to or called up by: DSK1.NAME (or DSK2, etc) whether from the keyboard or from within the Program or Loader.

Size==this indicates the number of sectors the file occupies on the disk, and can be a clue to the required Load procedure.

Type==tells what type of storage format was used in placing the File on disk. Usually the BEST CLUE to "What it is".

Rec==Length, in bytes, of the Records in the various File Types:

Display/Variable: "D/V 80" = 80 bytes Maximum per Record in the File.

Display/Fixed: "D/F 80" = 80 bytes on Each Record in the File.

Internal/Variable "I/V 254"

Internal/Fixed "I/F 128"

Program "PROG" = (no certain Length or Record definition.)

P==Protected (supposedly) from being copied or written to, by a rather simple scheme. It does have its uses, however.

C. FILES found during the cataloging of a disk will depend on the type of Program or Programs on the disk.

1. If it is a disk of "I-Basic Games", you will find mostly "Program" type Files and possibly some "Int/Var 254" type. You may find ONE File named "LOAD", and possibly one or several "Dis/Fix 80". There may be a "Dis/Var 80" named "READ-ME", but it is unlikely you will find a large file space of D/V 80 Files.

2. A disk dedicated to a single Program of large scope, such as Funnelweb, will usually have a large amount of documentation in D/V 80 files, and possibly a "Reader Program" to run in I-Basic. If the Program is made to run WITH the I-Basic module, it will almost always have a "LOAD" file and multiple files of "PROGRAM" or D/V 80 types which the "LOAD" File accesses and Loads.

3. A disk which has a number of Display/Fixed 80 Files and NO Program named "LOAD" is probably Machine Language (Assembler) Files which will have to be LOADED and RUN from Editor/Assembler Option 3 or Funnelweb Loader 4, particularly if the Files are named sequentially: "GAME", "GAME", "GAME", etc.

Other disks may have a series of "Program" format Files which will not Load into I-Basic. These will probably have to Load via E/A Option 5, ("RUN PROGRAM"), or Funnelweb Loader #3, or TI-Writer Option #3, particularly if there are names such as "UTIL1", or a series of sequential names.

4. A disk full of text files will USUALLY be "Dis/Var 80" Files, since that is the Form that TI-Writer uses, but other word processor programs operate with some variations of D/V 80, or other File Formats, so you may find other "Text" file types, with possibly a File Reader Program. Within specific programs there may also be Text Files which are NOT D/V 80. SOURCE Code Files of Assembler Programs are also Dis/Var 80 Files, so technically, THOSE "Text" files could be called in-completely processed "Programs".

5. A disk with Dis/Fix 128 Files usually either has been downloaded from some other system, and could be an ALE Graphics File, or it has been ARCHIVED. These Files must be De-archived and returned to their original form before the actual File types enclosed in the archive can be determined.

## QB MONITOR ~ QB-99 ~ NEWSLETTER

Reprinted From HOUSTON User Group (HUG) Newsletter

### PROGRAMS, FILES, AND LOADING.....Page 2

6. Some other file types such as Internal/Variable 128 may be used by Graphics programs or other DATA programs. The source Program may be necessary for these Files to work. Archived Files which have also been COMPRESSED are also in Int/Fixed 128 format.

7. ALL types of Files may show up being used for DATA Files, depending on the Disk and Program involved. Int/Var 64 and Int/Var 128 seem to be common in this form.

Any Int/Var 254 Files of LESS THAN 45 sectors will NOT be I-Basic Programs, but rather Data or Text Files.

#### D. LOADING PROGRAMS

1. The easiest programs to Load are those which are in module form, in which case the Module Menu is usually close to self explanatory, but the "user's guide" may tell you a lot more about HOW TO use it best or easiest. Some info and tips may not even be in the module guide either, for that matter.

2. The next easiest to Load (not necessarily to USE) are the Programs which combine a Module with a Disk. Usually the Module has a menu screen and then auto-loads the portion of the disk which is being used. The documentation may be on the disk but usually is in a users guide and ranges from very sketchy ("hints") to downright over-whelming (the manual for TI-Writer, for instance).

3. Cassette was the first storage media for most home computers, 99/4A included. Cassette-based programs are virtually ALL RUNable programs and, for the 99/4A, usually RUN in BASIC, since about the time people started widely using Extended Basic they also went to disks. Loading is by "OLD CSI" for both BASIC and I-Basic, as per the Users Reference Guide (green book) pp 11-49 to 11-52. See also the previous article on TI-Basics for Loading information.

4. A. RUNable "Programs" (for Basic/I-Basic) on disks are usually "PROGRAM" format up to about 40 sectors, or, for I-Basic Programs longer than 40 Sectors, "Int/Var 254". These are loaded and run via "OLD DSK:filename", and then "RUN", or for I-Basic, via "RUN "DSK:filename" which loads and starts RUNNING immediately. Maximum size of Int/Var-254 I-Basic Files is 97 sectors, and they only exist in Memory Expansion systems.

B. Following are Tips to determine Basic/I-Basic type, since Basic/I-Basic will each Load the other type of

Programs from cassette or disk, but not necessarily RUN properly:

Most BASIC Programs will load and RUN in I-Basic, but may get a "BAD VALUE .." error if they use characters from set 15 or 16, or if multiple colons were used in PRINT lines for Linefeeds. I-Basic Programs not only will not RUN in BASIC, they will not even LIST properly. SO if you are in BASIC and try to LIST the program and it goes real slow and has a lot of garbage, it is I-Basic. IF it LISTS ok in BASIC and has ALL single-step LINES (no double colons between statements in a LINE) and NO statements such as "DISPLAY AT"; "LINPUT"; or anything whatever to do with SFRITES; then it probably is in BASIC.

C. Another (sub-routine or partial Program) form used with I-Basic Programs is the "MERGE" format: Dis/Var 163. These can be loaded via: MERGE "DSK:filename", to combine with the Program already in memory, and then RUN. Any regular Program which is saved via SAVE DSK:filename, MERGE will be stored in this MERGEable Format--Dis/Var 163, and must be re-loaded via the MERGE command.

D. ANY I-Basic PROGRAM File with name "LOAD" will auto-load and RUN if it is located on DSK1, when the I-Basic module is first selected from the console menu. These programs are usually Entry or Menu Programs for a larger Program or group of Programs on a disk. Only ONE "LOAD" file can exist on a disk at a time.

E. NOTE: PROGRAM format files that CANNOT be loaded and RUN are created by Scott Adams Adventures (usually 54 sectors); by the Personal Record Keeping and other modules as data files, in non-determinant lengths, and by the Tunnels of Doom Program, (usually 52 sectors). TI-Artist creates two 25-sector PROGRAM files for each picture SAVED, with names tagged /P and /C. Such "Program" files can ONLY be re-loaded and used with the module or Program which created them, usually. Other non-executable PROGRAM files exist as DATA Files, such as CHARAI in many Programs.

ANY "PROGRAM" format files over 49 sectors will NOT be BASIC / I-BASIC; it is probably a Data File.

F. Additionally, the I/V 254 and the D/V 163 formats MAY be used for DATA files in some programs, and these would NOT be Files which could be Loaded and RUN, regardless of length.

G. Generally, for PROGRAM files (and Int/Var 254 files) which appear by CONTEXT of the disk or FILENAME to be RUNable Programs files, try I-Basic

"OLD..", then if they will not load, try using E/A S to Load them. If they STILL will not Load, they may be DATA files.

HINT: Funnelweb SHOW DIRECTORY will indicate whether a "Program" File is ID or EA type, also; if you press " = " while on the directory screen, FM 4.0 will check EACH "Program" File and indicate the type in the REC column by these initials. REC will remain blank for any "Program" file which is a DATA File. This is one of the best ways to find out WHAT IT IS, for the required operating mode of a "Program" File.

S. A. MERRY-IMAGE Programs are "Program" files of 33 sectors (or less) and usually have "chained" names like "UTIL1"; "UTIL2"; "UTIL3". They are NOT RUNable in B/I-B. These are Machine Language Program Image files and are Loaded via the Editor/Assembler option 5("Run Program"), TI-Writer Option 3 "Utility", or Funnelweb Loader 3 (in Formatter Menu Screen, Loaders Option). These Files will give an "I/O ERROR 50" if you attempt to Load them via I-Basic. These files are actually the series of numeric values which comprise the Machine Language program in the memory of the computer, and are created by the Ed/Assembler from Object Code files by using the SAVE Utility. They can be no more than 8192 bytes in length, therefore will catalog at 33 sectors (or less). When they are Loaded, they are passed directly from the Device where they are stored into the memory, in sequential order, and therefore Load relatively quickly.

B. Memory Image files are loaded based on a three "Word" Header at the beginning of the File, as follows:

1. The first Word is a flag which indicates whether this File is the Last File in a series by value zero; any other value indicates that there are more Files to Load in the series, based on the sequential names.

2. Second Word (third and fourth bytes) of the File is the Length of the Memory Image File, in bytes, including the six bytes of the Header.

3. Third Word in the Header is the CPU Address to begin Loading this File; the beginning address in memory. This means that each 8k (or less) block can be Loaded to an individual area of memory. EXECUTION (at CPU level) begins automatically at the first byte of the first File block loaded, after the Last File has been Loaded.

## QB MONITOR ~ QB-99'er NEWSLETTER

Reprinted From HOUSTON User Group (HUG) Newsletter

PROGRAMS, FILES, AND LOADING.....Page 3

C. LOADING a Memory Image File series can be done from Ed/Assembler Option 3, by entering the device and name of the FIRST File of the series:

DSK1.GAME1 (enter)

and the Loader will Load this File, search for and Load the File GAME2 if it is present, etc, and then Auto-Start the Program.

The default Filename (not shown on the screen!) is "DSK1.UTIL1", so pressing Enter at the prompt will attempt to load UTIL1, UTIL2, etc if they are on the disk. TI-Writer Option 3, "Utility" and Funnelweb Loader 3 operate in the same way. It might be noted here that the TI-Writer module is simply a Memory Image Loader with "EDITA1", "EDITA2", as the default names for the Files if Editor is chosen, and "FORMA1", "FORMA2"....as the names for the Files if the Formatter is chosen. Therefore, if you re-name the DN1000 files MGR1 and MGR2 to FORMA1 and FORMA2, you can Load DN1000 by choosing the "Formatter" (Option 2) on the TI-Writer module, as an example.

The "Utility" or "Run Program" choices on each Loader allow a different Filename or device to be entered, over-riding the default names, to allow you to Load the actual Filenames from whatever disk the Files are located on. In fact, E/A supposedly will Load these Files from Cassette, if you can get them ONTO cassette.

6. A. OBJECT CODE Files are Assembly Language Program Files in Display/Fixed 80 format which have been created with a Assembler from a SOURCE CODE listing, usually via the Editor/Assembler, but possibly from another compiler such as C-compiler or Fortran, etc. These files are of two types:

Un-compressed Tagged Object Code--which is in Hexa-decimal form for easier re-editing, but must be converted to machine language bytes DURING LOADING, and thus Loads more slowly. These may be up to 185 sectors.

Compressed Tagged Object Code--which has been converted to machine language bytes, and occupies only about one-half as much disk space, and Loads faster. Note: there is no flag on the file to indicate

the distinction between Compressed and Un-compressed Files, but if viewed with a Sector Editor program, the Compressed cannot be read. These Files may be UP TO 185 sectors in length.

B. Object Code Files can be either:

---Relocatable, for which the Loader (Module or Program) may place the Code in different areas of memory and then establish entry points by Reference, and

---Absolute, which must be loaded into a specific location in memory each time.

C. LOADING Tagged Object Code may be done via E/A Option 3, Funnelweb Loader Option 4, or Mini-Memory Option 1, "Load and Run"; for ALL forms of Object Code. The procedure is as:

1. Choose E/A 03 or FM (Loader Menu) 04 or M/M 81;

2. At the prompt "FILENAME?", enter the Source and Filename(s): (there is no Default Name built in) DSK1.FILE1 (enter)

3. Loader will prompt for "FILENAME?" again:

(assuming a FILE-DSK1.FILE2 (enter) (NAMEs series are (present on the DSK1.FILE3 (enter) (disk.

4. Then, if no more Files are required, (enter);

5. Loader will ask for "PROGRAM NAME?" (unless the last File AUTOSTARTED): You must Enter the Name assigned as the Start of Program REFERENCE. This REF may be listed in the final sectors of the last File, if you do not know what it is. ALSO, Funnelweb will list out the REF table of names for you, if the Program did not Auto-start, so you can look for something like: "START", "GO", "RUNIT", "BEGIN", (the Program name), or some likely REF Name of six digits or less, to try as a "PROGRAM NAME".

D. It is also possible to Load Hexa-decimal form Object Code via the Extended Basic Module by using the following series of commands:

```
CALL INIT
CALL LOAD("DSK1.filename")
CALL LINK("program name")
```

EXCEPT: Compressed Object Code cannot be Loaded through I-Basic DIRECTLY.

NOTE: I-B only recognizes the Low Memory area >2000 to >4000, and provides reduced support routines, as compared to Code Loaded within the E/A or Mini-Memory environments. Also, I-Basic will not resolve external REFERENCES between Files and requires EQUates into the BRON utilities after CALL LINK, rather than accessing them directly. Therefore, it is possible to Load a Program which will ERROR out during running on an incomplete REF-DEF series or lack of GPL Utility support via EQUates into BRON. Also, it is a SLOW Loader due to GPL operation basis. However, there ARE Object Code Files which have been designed to Load and Run via the I-Basic module, and they DO enhance performance of an otherwise SLOW "Basic" into something like machine language speed.

Although NO DIRECT METHOD of Loading Memory Image (E/A SAVE) Files into Extended Basic is provided for, Extended Basic IS used as a Machine Language File Loader by means of Auto-RUNning Files named LOAD which become Assembly or Program Image file Loaders, or which are HYBRIDS and contain Machine Language sections "hidden" within what the 99/4A considers to be the I-Basic Program, either directly or by CALL LOADs from DATA statements. The File "LOAD" in Funnelweb, which Auto-RUNS in I-Basic, is one such "Hybrid" Program, with over 25 sectors of Machine Language Code.

Obviously, attempts at EDITING such a "I-BASIC" PROGRAM can be disastrous since it constitutes an Absolute form of Memory Image Program, at least in part.

E. Another possible way to accomplish a "Load and Run" of Object Code Files is by BASIC, with the Editor/Assembler module or the Mini-Memory module in place, which adds such subprograms as CALL LOAD and CALL LINK to BASIC. Again, ALL system Utilities and REFs must be established, via CALLS and LOADING separate Files.

F. I hope that all of this does not leave you more confused than ever. If this is any help to you or if you spot an error or something to add, please write and tell me:

R. Luepkin, c/o Houston U.G. (JUN88/PP)



Great Lakes Computer Group, Inc.

MAY 1989

PAGE 1

VOL. 7, NO. 12

### II-WRITER TIPS #1 - by Bob Seddon -

#### EDITOR MARGINS VERSUS FORMATTER MARGINS

There are two kinds of margins in II-Writer: Editor Tabs & Formatter Dot commands.

#### EDITOR TABS

When you create text with the Editor you use margins called Tabs. Tabs are set via CTRL c (PROMPTS), t (Tabs), Enter. This sequence of keystrokes makes the Tab Line appear across the top of the screen. You can type over the Default settings at 0 and 79 and reposition L and R to make on-screen tabs any width within that range.

#### PRINTING WITH THE EDITOR

You can print text created in the Editor with the Editor itself by CTRL c (PROMPTS), f (FILES), pf (Print File), Enter. This sequence of keystrokes prints text with margins equal to the tab settings; the printer output resembles the screen.

However, there are advantages in NOT using this method to print. If you use the Formatter instead of the Editor you can print lines longer than 80 spaces. You can automatically number successive pages. You can put Headers at the top of each page, Footers at each bottom. You can make the R margin flush with the .FI;AD command. You might want to use the Ampersand to underline, the Each to Print Bold. You can double space and set page length.

#### PRINTING WITH THE FORMATTER

To print through the Formatter you must Save the file, Exit the Editor, Load the Formatter, reload the file, and then print. The Formatter prints the file according to the Dot Command instructions.

#### FORMATTER DOT COMMANDS

Formatter margins are also set in the Editor, but not the same way as the Tabs. Instead, Formatter margins are typed in (usually on line 0001) as Dot

Commands (.LM n;RM n). Dot Command margins (if present) override Tab margins when text is printed through the Formatter. If there are no Dot Commands the file will print out according to the Tab margins. The Formatter follows the Dot Command instructions but does not print the Dot Commands as it does text. The Editor, on the other hand, not only ignores Dot Commands but also prints them just as it will any other text, since it cannot make the distinction between Dot Commands and regular text.

Quite often you will want to print your text with margins EXACTLY the same as on-screen. There are at least two reasons you might want to do this:

#### (1) HYPHENS

If you pack in as much text per line as possible you will want to break words and hyphenate them; if you do so, the final printed output must break the words at the same place you did. Otherwise, your text will take on the appearance of this particu- lar sentence.

#### (2) MULTIPLE COLUMNS

If you create text with narrow columns so that you can put several parallel columns on one page you need to count the EXACT number of lines. Line numbers down the left column give you this number (minus the lines devoted to printer commands) if your on-screen equals your printed work. (NOTE: see box at end of article about the advantages of narrow columns.)

There is a trick you must use to make your on-screen work created in the Editor resemble the printed output of the Formatter so that each resembles the other line-by-line.

#### R TAB ONE / HIGHER THAN .RM

If you use the Formatter you must set the R Tab one digit higher than the setting of the .RM dot command. The difference between the Formatter and Editor is that the Formatter will print ON the .RM column. The Editor prints UP TO (but not on) the R Tab.

TAB SETTINGS: 0 & 31  
SPACES USED ON-SCREEN: 0 - 30  
(31 ACTUAL SPACES OCCUPIED)

DOT SETTINGS: 0 & 30  
COLUMNS PRINTED ON: 0 - 30  
(31 ACTUAL SPACES OCCUPIED)

In this article I set the Editor margins at 0 and 31 to fit three columns on the page. The Formatter settings are at .LM 0;RM 30. You can see that the columns printed at a width of 31, not 30.

123456789 123456789 123456789  
L....T....T....T....T....T....T

When counting, remember to begin ON zero: call the 0 a 1, 1 a 2, 2 a 3, etc. A follows the last T. We cannot print it here for the very reason being discussed! Counting the number of spaces used in each line on printed work is not difficult. Nor is it hard to count spaces used on screen. It is even easy to memorize the rule for making Editor margin width one character longer than dot command margin width. The one frustrating thing about this whole business is allowance for a L margin on zero!

Instead of making .RM one digit smaller than the R tab setting you have the option of achieving the same effect by making the .LM one digit greater than the L Tab of the Editor. The option of using zero as a L Tab adds confusion to this issue because you can also set .LM at zero; thus, this tactic is of no particular benefit for you.

#### SPACE(S) AFTER . ; ? ! REFORMAT VERSUS .FI

When you Reformat, the Editor packs in text according to its own set of rules, rules different from the Formatter .FI command. If you leave only one space after a period, the Editor's Reformat command will NOT increase the spacing to two places. The Formatter, on the other hand, ALWAYS leaves two spaces after periods, whether you want it to (at sentence ends) or not (after initials).

We can prevent the Formatter from increasing the single space after the final dots of initials, abbreviations, etc. by putting a carat between such dots and the next letter.

Similarly, we need to force in two places after ;, ?, and !. If you merely leave two spaces after each one the Formatter will reduce your two spaces after each of these down to one unless you follow them with a carat, then the space. Optionally, you can key in two carats (and no space).

SPACES LEFT AFTER:			
	period	initial	?!:
Editor	1	1	1
Reformat			
Formatter	2	2	1
.FI			
Remedy	space twice	carat once	carat twice

**FORCED IN CARRIAGE RETURN**

The Formatter makes a decision to Wrap based on the R tab setting and whether a word (or ANY group of characters) occupies or exceeds that setting. Usually the last PRINTED character in a paragraph is a period and if it falls on the last occupiable position (R tab setting minus one) you must be careful where you place the carriage return.

- (1) If you space once after the period, then Key CTRL M, there is no problem.
- (2) If you cursor down below your text, then Key CTRL M, there is no problem.
- (3) If you key CTRL M in the position directly following the period, the last word in a paragraph will not fit at the end of the line and will drop to the next line.

When it drops, you notice that it SHOULD fit, even when you account for the space before the word and the period following. The Editor Wraps the word around to the next line because it treats the carriage return

following the period as part of the word, even though the carriage return is not a printed symbol.

If this happens to you, you must break the text after the period and before the carriage return (CTRL G), then Reformat. The word will now NOT wrap to the bottom line. The carriage return also moves up to the original line.

**FORCED IN FORMATTER COMMAND**

A similar problem occurs when you precede a word with an ampersand, carat, or @. Let us consider the ampersand which is, of course, a formatter command to underline any word it precedes. The Formatter .FI Command ignores the ampersand and packs in Text as though the ampersand were not there. Unfortunately, the Editor treats the ampersand as a regular character when Reformatting and, so, will make a decision to Wrap a line based on the presence of it within a line of text, just like the carriage return. This anomaly makes it difficult to create a line of text which appears on screen exactly as it will print.

There is a technique to insert these codes in front of any (or even every) word on the line. Unlike the carriage return which FOLLOWS a word, a Formatter command PRECEDES it, making the previous technique impossible. Turn off wordwrap with CTRL O. This turns your cursor into a hollow rectangle. Move the cursor to the letter before which you want an &. Key in Insert (FCTN 2). Key in the ampersand. Everything right of it will move right one column. This is the only way you can make a character appear ON column R.

You can only insert one such ampersand per line using this trick UNLESS YOU RESET THE EDITOR R MARGIN TO A HIGHER NUMBER. If you insert an additional character anywhere else on the line and do not first increase the R margin the last character on the line will vanish. You can precede EVERY

word in the line with a non-printable character so long as you increase the R Tab enough so that all text and all codes fit on that line. The only restrictions which apply are that you may not mix text and code such that you exceed 80 spaces; nor can you Reformat afterwards.

**THE NARROW COLUMN ADVANTAGE: NO WINDOWING**

I find it convenient to set on-screen margins so I can see all text without Windowing left and right. Since I also like to leave the four-digit column numbers on the left side of the screen at all times, the highest possible R margin setting is 34. (34 is off screen, but Wordwrap causes text to occupy 33 by briefly Windowing right, then left, when you type on 33 itself.)

You can turn off the column numbers (FCTN 0) to see six more columns of text (4 digits and 2 spaces) which lets you set the margins at 0 and 39. You can even set R at 40 (which is off screen) so that Wordwrap will cause text to occupy 39; this makes a REAL 40 column screen. Naturally, the R Tab can be reset anytime to Reformat to any desired margin width up to the on-screen maximum of 0-79.

**79 COLUMN SCREEN**

I am sorry to break the news to you that you only have a 79 column screen, not the 80 column screen you thought you did! When in Wordwrap the Editor does not let text occupy the column of the R margin (on screen or when printed with the Editor). If you create text with Wordwrap on, the Editor alone cannot print 80 columns: its extremes are 0 and 79, and 79 is not printed on. The arithmetic is tricky because of the presence of the zero. If your Editor margins are on 0 and 79 you can only print 80 column text IF your dot commands are set at 0 and 79! AND IF you use .FI; nor will on-screen equal output.

### II-WRITER TIPS #1 - by Bob Seddon -

#### EDITOR MARGINS VERSUS FORMATTER MARGINS

There are two kinds of margins in II-Writer: Editor Tabs & Formatter Dot commands.

#### EDITOR TABS

When you create text with the Editor you use margins called Tabs. Tabs are set via CTRL c (PROMPTS), t (Tabs), Enter. This sequence of keystrokes makes the Tab Line appear across the top of the screen. You can type over the Default settings at 0 and 79 and reposition L and R to make on-screen tabs any width within that range.

#### PRINTING WITH THE EDITOR

You can print text created in the Editor with the Editor itself by CTRL c (PROMPTS), f (FILES), pf (Print File), Enter. This sequence of keystrokes prints text with margins equal to the tab settings; the printer output resembles the screen.

However, there are advantages in NOT using this method to print. If you use the Formatter instead of the Editor you can print lines longer than 80 spaces. You can automatically number successive pages. You can put Headers at the top of each page, Footers at each bottom. You can make the R margin flush with the .FI;AD command. You might want to use the Ampersand to underline, the Each to Print Bold. You can double space and set page length.

#### PRINTING WITH THE FORMATTER

To print through the Formatter you must Save the file, Exit the Editor, Load the Formatter, reload the file, and then print. The Formatter prints the file according to the Dot Command instructions.

#### FORMATTER DOT COMMANDS

Formatter margins are also set in the Editor, but not the same way as the Tabs. Instead, Formatter margins are typed in (usually on line 0001) as Dot

Commands (.LM n;RM n). Dot Command margins (if present) override Tab margins when text is printed through the Formatter. If there are no Dot Commands the file will print out according to the Tab margins. The Formatter follows the Dot Command instructions but does not print the Dot Commands as it does text. The Editor, on the other hand, not only ignores Dot Commands but also prints them just as it will any other text, since it cannot make the distinction between Dot Commands and regular text.

Quite often you will want to print your text with margins EXACTLY the same as on-screen. There are at least two reasons you might want to do this:

#### (1) HYPHENS

If you pack in as much text per line as possible you will want to break words and hyphenate them; if you do so, the final printed output must break the words at the same place you did. Other-wise, your text will take on the appearance of this particu- lar sentence.

#### (2) MULTIPLE COLUMNS

If you create text with narrow columns so that you can put several parallel columns on one page you need to count the EXACT number of lines. Line numbers down the left column give you this number (minus the lines devoted to printer commands) if your on-screen equals your printed work. (NOTE: see box at end of article about the advantages of narrow columns.)

There is a trick you must use to make your on-screen work created in the Editor resemble the printed output of the Formatter so that each resembles the other line-by-line.

#### R TAB ONE # HIGHER THAN .RM

If you use the Formatter you must set the R Tab one digit higher than the setting of the .RM dot command. The difference between the Formatter and Editor is that the Formatter will print ON the .RM column. The Editor prints UP TO (but not on) the R Tab.

TAB SETTINGS: 0 & 31  
SPACES USED ON-SCREEN: 0 - 30  
(31 ACTUAL SPACES OCCUPIED)

DOT SETTINGS: 0 & 30  
COLUMNS PRINTED ON: 0 - 30  
(31 ACTUAL SPACES OCCUPIED)

In this article I set the Editor margins at 0 and 31 to fit three columns on the page. The Formatter settings are at .LM 0;RM 30. You can see that the columns printed at a width of 31, not 30.

123456789 123456789 123456789  
L....T....T....T....T....T....T

When counting, remember to begin ON zero: call the 0 a 1, 1 a 2, 2 a 3, etc. A follows the last T. We cannot print it here for the very reason being discussed! Counting the number of spaces used in each line on printed work is not difficult. Nor is it hard to count spaces used on screen. It is even easy to memorize the rule for making Editor margin width one character longer than dot command margin width. The one frustrating thing about this whole business is allowance for a L margin on zero!

Instead of making .RM one digit smaller than the R tab setting you have the option of achieving the same effect by making the .LM one digit greater than the L Tab of the Editor. The option of using zero as a L Tab adds confusion to this issue because you can also set .LM at zero; thus, this tactic is of no particular benefit for you.

#### SPACE(S) AFTER . : ? ! REFORMAT VERSUS .FI

When you Reformat, the Editor packs in text according to its own set of rules, rules different from the Formatter .FI command. If you leave only one space after a period, the Editor's Reformat command will NOT increase the spacing to two places. The Formatter, on the other hand, ALWAYS leaves two spaces after periods, whether you want it to (at sentence ends) or not (after initials).