





JUNE

1988

HUG TIBBS - 24-hour BULLETIN BOARD (713) 781-4844 300/1200 bps

MEETING SCHEDULE
FIRST SUNDAY OF EVERY MONTH
(2nd Sunday if 1st Sunday
is on a holiday weekend)

The Houston Users Group is organized to promote the use of the TI-99/4A Computer. We invite persons of similar interest to join with us. HUG is not affiliated with TEXAS INSTRUMENTS, INC.

AT THE NEXT MEETING

SUNDAY, JUNE 5, 1988 2:00 P.M.

St. John's School - 2401 Claremont

AGENDA for JUNE MEETING: TELCO demonstration by Mike Connell will be attempted with a temporary phone line in place (we hope).

IN THIS ISSUE

Articles:

TI Writer Tips #2. by Bob Seddon Funnelweb 4.0 Hints/Features April's Library Addendum

TI Faire Recap

Back to Basics--Perspective and Cassette Long Files

Files, Programs, and "Loading the Unknown"

Other Departments: Editors.Echoes HUG Classifieds

Meeting and Member Info Printer Commands Update

1988 HUG OFFICERS

THIS NEWSLETTER IS PUBLISHED MONTHLY BY THE HOUSTON USERS' GROUP, JAJ "HUS". ANY OPINIONS OR ENDORSEMENTS EXPRESSED IN ARTICLES ARE THOSE OF THE INDIVIDUAL AUTHORS AND ARE NOT NECESSARILY THOSE OF THE HOUSTON USERS BROUP, ITS OFFICERS, OR ITS MEMBERS. REFERENCE TO A PRODUCT OR COMPANY IS NOT AN ENDORSEMENT OF THAT ITEM.

000000000000000000000000000000000000000								000	000000000000000000000000000000000000000							
0)		1 '	9	8	8		0	0		1	9 :	9	₿ •		B
000000000000000000000000000000000000000											00					
0				JUN	E			0			1	JUL	Y			B
0								0								ø
0	S	Ħ	- T	ķ	T	F	S	0	S	M	Ţ	₩	T	F	S	B
O	-	-	-	-	-	-	-	Q	-	-	-	-	-	-	-	0
٥				1	2	3	4	0						i	2	a
0	(5)	6	7	В	9	10	11	0	3	4	5	6	7	8	9	0
0	12	13	14	15	16	17	18	Q	(10)	11	12	13	14	15	16	0
O	19	20	21	22	23	24	25	0	17	18	19	20	21	22	23	0
0	26	27	28	29	30			Đ	24	25	26	27	28	29	30	Ö
Đ								0 (31							٥
000000000000000000000000000000000000000)							

Note: July, August, September, and November meetings will be on SECOND Sundays....

HUG MEMBERSHIP INFORMATION

Annual membership fees for HUG are \$15. A ONE TIME initiation fee, for new members ONLY, is \$10. initiation fee, used to maintain user group services. entitles the new member to receive six free programs - of his own choosing - from the groups Public Domain and Shareware software library. Members receive a monthly issue of the HUG newsletter, may order programs from the software library and are permitted access to all "user" levels on the Group's bulletin board (BBS), HUG-TIBBS. All renewal and/or new membership fees must be sent to the Group's Secretary at: Houston User Group, c/o Ruth Sandy Herman, 6219 Sanford Road, Houston, Tx. 77096. Membership applications are available either at the same address, at meetings, or from the (BBS): HUG-TIBBS at (713) 481-4844. Please direct any questions regarding current membership to the Membership VP. Kim Peterson (713) 498-3047.

For a free sample of this Newsletter please contact the Editor. Back issues of the HUG Newsletter are available at \$1.00 (postpaid)/issue, or \$.50 / issue at the meetings, for members. Call editor for availability.

HUG MEETING LOCATION MAP

	← CLARENONT						
1	27. HOOL . T2						
WESTHEINER RAI.							
RICHIONO AVE.	KEHEY ->						
SIE FRUY (577)							
	C-BUFFALO SPEEDURY						

Commercial A	dvertising R	Individual HUG Members				
Full Page:	\$15.00 per 10.00 *	AD.	List your PERSONAL software/hardware items			
1/4 * 1/8 *	7.50 "		wanted or for sale for			
Business Cards	2.00 "	ıi	FREE			
(Multi-Mo. d:	isc. availabl	in this section. Call Editor: 469-5089				

FOR SALE

TELECOMMUNICATING EQUIPMENT - Compuserve Starter Kit unopened \$25.00, Volksmodem 300 Baud MODEM, ac adapter, TI RS232 cable.... \$40.00 Call RON HODGMAN (713) 358-8618 evenings.

BASIC EQUIPMENT - TI-99/4A console (still in box), XB, Speech Synthesizer. \$75.00 Leave Msg on HUG-TIBBS to: DAVID HIRSCH

SYSTEM - Console, PEB w/ TI-9S drive, TI-Disc Controller, RS/232, 32 k card, plus stand-alone disc controller and double box w/ two SS TI-drives, Extra console, Widget, X-Basic; TI-W; MPlan; TE-II; PRK; HBN; Navarone Data Base Man.; power strip, discs--\$350.00. Also, 13" Color Monitor--\$100.00. Epson (TI) NX-80 Impact Printer--\$100.00. Also has for sale separately a Brother 50 typewriter with 2k interface. Contact James E Keith of Johnson Users Group, at 713-333-4504.

NON-MEMBER has: Console, PED w/ Drive, TI controller, R9232 and 32k cards, plus Extra Console (new in box); External Drive in ext. box; Speech Synth.; Printer; TJ-Writer, Multiplan, X-Basic. Call Irving Feld at 723-9898.

INFORMATION AND DRAWINGS - on John Willforth's "32k in the console" micro expansion can be obtained by sending your address and Self-addressed legal size envelope to:

Editor, HUB News; Box 690311, Houston Texas 77269.
This is the console mod which uses 4 chips 6264LP15 to replace the 32k card in PEB. Costs (\$20.00 and MORKS just like the original memory. Great for "second console" or for non-PEB-systems, or if your 32k card dies! Requires console dis-assembly and soldering, and about 6 hours. Plans courtesy of J. Willforth and other supporters of the 99/4A.

SUBMITTALS of articles of general interest are welcomed, with implicit under-standing of permission to copy and publish these documents. It is requested that all submittals be in publishable format if typed, or in DV/80 disk files. Braphics, art, schematics, etc, should be submitted in final form and size, and repro quality. Other information and articles are welcomed by the Editor as data sources, and may be edited and incorporated as needed.

Please indicate Original Source and publishing sources of information--Copyright/Reserved material cannot be used without author's MRITTEN permission. Thank you, (Ed).

TI-WRITER TIPS #2 - by Bob Seddon -

DUPLICATE KEYBOARD COMMANDS

COMMAND/ESC CTAL C. FCTN 9 (also) FCTN # DEL. CHAR GTRL F. FCTN 1 DEL. LINE CTRL n, FCTN 3 INSERT LINE CTRL o, FCTN 8 DOWN ARROW CTRL x, FCTN x CTRL s, FCTN s LEFT ARROW RIGHT ARROW CTRL d. FCTN d UP ARROW CTRL . FCTN . INSERT CTRL g. FCTN 2 REFORMAT CTRL r, CTRL 2 NEW PAGE CTRL p, CTRL 9 CTRL M, CTRL 6 CTRL 8, CTRL 5 CTRL 4, CTRL 3 NEW PARA LAST DARA NEXT PARA ROLL DOWN CTRL &, FCTN 4 ROLL UP CTRL b, FCTN 6 TADa CTRL 1. FCTN 7 WORD TAB CTRL W. CTRL 7 00PS! CTRL z, CTRL 1

The above chart lists preferred alternatives first, on the left. You have a choice with duplicates and in some cases the choice is very, very clear. In some cases both alternatives are bad; in some, both are okay, or at least, acceptable. Some are situational, i.e., the strokes preceding or following will determine which is better; in other words, in a few special cases the right-hand alternative is better. I offer a couple of rules to help with the selection process:

- Anytime you can avoid using two hands you are better off, unless the onehanded alternative is awk ward.
- (2) If you do have to use two hands, avoid any move that makes use of the top row of keys (keys with numbers).
- (3) Paired moves ought to complement each other. By "paired" I mean Roll up/ then down, Last paragraph/ next paragraph, etc.
- (4) If no other insight is available, a move may be considered better if its mnemonic.
- (5) Some keyetrokee procede or follow others in a sequence of moves, moves which tend toward fluidity.

COMMAND/ESC (also) CTRL c, FCTN 9 FCTN =

rist of all, I object to the name selected by TI-Writer. It is confusing to use the term "escape" for anything other than Printer Control Code 27 (the escape command). Nor do I like the term "command". There are Direct commands (such as CTRL 3 for color) and there are Prompted commands which make use of a MASTER PROMPT LINE. This, in my opinion, ought to be called PROMPTS.

This is the unly command in TI-Writer which can be achieved in three different ways. The 3 commands do the same thing, whatever you choose to call them, but only if you are currently Editing.

CTRL c is the best choice for every occasion because FCTN 9 is a hard reach and because. FCTN = can accidently crose a file: if you have a file onscreen and call up the Disk Directory while in that file you can make a mistake when leaving the disk directory if you are in the habit of calling up PROMPTS with FCTN =.

The accident works like this: when you want to leave the disk directory and go back to your file you must key ENTER. Habit, though, causes you to call up the MASTER PROMPT LINE instead. If you habitually use CTRL c or FCTN 9 to call up the MASTER PROMPT LINE there is no problem, because the program will DECP (telling you that you made a mistake) and do nothing; now, alerted, you can key the correct response, ENTER.

If, on the other hand, you are in the habit of using FCTN = to call up the MASTER PROMPT LINE you will reap disasterous results. After reviewing the Disk Directory you are supposed to key FNTER to return to your file. If you key FCTN = you will Purge the file and return to the title screen.

The point is: you will never accidently Purge your file if you mistakenly key FCTN 9 or

CTRL c instead of ENTER, but you will accidently Purge your file if you mistakenly key FCTN = instead of ENTER. The solution is obvious: never, ever use FCTN = to call up the MASTER PROMPT LINE, because you may accidently Purge your file. Never use FCTN 9 because it is awkward. Always use CTRL c; it is an easy reach.

DEL. CHAR DEL. LINE INSCRT LINE

CTRL f, FCTN 1 CTRL n, FCTN 3 CTRL o, FCTN 8

e can reference Rule (2) for all three and Rule (1) for both DELETES. INSERT LINE would seem to be a one hander (FCTN 8), but many people are probably in the habit of keying 8 with the right hand; therefore, CTRL o is better. For some reason, TI did not print INSERT LINE on the Quick Reference Card which comes with TI-Writer.

CTRL f is a convenient reach:
left thumb and forefinger.
Also, after Deleting you can
Insert (CTRL g) with a very
small finger movement: (f and
g are adjacent keys.)

THE ARROW KEYS

DOWN ARROW LEFT ARROW RIGHT ARROW UP ARROW CTRL x, FCTN x CTRL s, FCTN s CTRL d, FCTN d CTRL e, FCTN e

Since most PROMPT LINE commands are CTRL key commands strongly recommend CTRL over FCTN for the arrow keys. We also cite Rule (1).

INSERT REFORMAT CTRL g, FCTN 2 CTRL r, CTRL 2

There is a special case for using FCTN 2 in the non-wordwrap mude, discussed several paragraphs below. Most of the time, though, you are in Wordwrap. When Wordwrap is on CTRL g is the better choice. We cite Rule (1) and (not so obviously) we also invoke Rule (5) with some complex logic.

INSERT is dependent not just on the immediately following move (typing in new text) nor even the second move (ARROW UP) but rather on the last move (REFOR-MAT). Rarely does REFORMAT not follow INSERT. The sequence of, moves is usually:

(1) INSERT

(2) Type in text

(3) ARROW UP

(4) REFORMAT

insert arrow reformat
FCTN 2, CTRL e, CTRL r
FCTN 2, CTRL e, CTRL 2
FCTN 2, FCTN e, CTRL r
FCTN 2, FCTN e, CTRL r
FCTR 2, FCTN e, CTRL 2
* CTRL g, CTRL e, CTRL 2
CTRL g, FCTN e, CTRL r
CTRL g, FCTN e, CTRL r

Please notice that there is a fluid sequence of moves for only one of the eight possibilities. This single instance uses only CTRL keys. We cancite Rules (1), (2), & (5). I recommend using CTRL g instead of FCTN 2 when breaking text. The deciding factor in selecting CTRL g over FCTN2 is the second move (REFORMAT), not the first move (ARROW UP). CTRL r is faster than CTRL 2.

As promised in the introductory paragraph on INSERT, there is a unique situation which calls for preferred use of FCTN 2 over CTRL q. Let us say you wish to amend a chart with Wordwrap off (CTRL 0; hollow cursor) such that you wish to INSERT a carat (SHFT 6) in the same column on every line. All text right of each carat on each line moves right one space. The fastest way to do this is to position the cursor on the first line, on that column. With Wordwrap off, you do a little dance:

(1) FCTN 2 [R thumb, L index]
(2) SHFT 6 [R index, L index]
(3) FCTN x [R thumb, L index]
(4) FCTN s [R thumb, L index]
(5) FCTN 2 [R thumb, L index]
repeat 2-5, 2-5, 2-5, etc.

Keep in mind the device of the Little Dance. It shows up in other forms again and again in Word Processing and is a very useful concept. NEW PAGE CTRL p, CTRL 9 NEW PARA CTRL m, CTRL 8

Rule (2) applies to both. We can cite Rule (4) if you can remember p stands for Page, not Paragraph.

LAST PARA CTRL 6, CTRL h NEXT PARA CTRL 4, CTRL j

ast and Next Paragraph are used in pairs. There are a lot of pros and cons here. Keys in and jare next to each other, making them easy to remember, but requiring you to use two hands. A case FOR using h and j is your ability to think in terms of h being fore" j; h "Heads for Home". While Roll Up/Down (FCTN 6/4) are two handers, CTRL 6/4 are not, even if (properly speaking) 6 is a right hand key which you probably will press with your left hand. 4 and 6 are not adjacent, are on the top row, and most commands are via CTRL key anyway.

I can think of one occasion when it would be better to use CTRL 6 and UTRL 4 in preference to the FCTN key counterparts. When reformatting a whole file to new margins it is best to begin at the top and do a little dance: CTRL r, CTRL 4, CTRL r, etc. For this reason, I break Rule (2) and use CTRL 6/4 in all cases.

ROLL DOWN CTRL a, FCTN 4
ROLL UP CTRL b, FCTN 6

R oll Up/Roll Down are used as pairs. Because they are one handers, CTRL a and CTRL b are the best choices. You can poise your hand like a spider: thumb on CTRL, little finger on a, and index finger on b. It is fairly natural to dance your digits back and forth as you read Up and Down. b is an easy mnemonic for "back". You may be tempted to think backwards that A comes "before" B.

The case against FCTN 4 and FCTN 6 is about the same as

with LAST/NEXT PARA. The case for 4 and 6 is now bolstered by the fact that Rolling Up/ Down and Paragraphing Up/Down are very similar operations, one pair via CTRL, the other pair via FCTN. Furthermore. for the CTRL key mnemonics inconsistent, alternates are with b (the second letter) taking you "back" and h (the first taking you "home". letter) With both Paragraphing and Rolling the number logic is backwards: 6 takes you Back: 4 takes you forward. I still favor using 4 and 6 for Paragraphing, a and b for Rolling.

TAB CTRL i, FCTN 7
BACK TAB CTRL t
WORD TAB ÇTRL w, CTRL 7

B ack Tab is not a duplicate command, but I included it because it is used in conjunction with Tab, which is. The hest way to use TAB/BACK TAB is CTRL t (left hand forefinger on the t) and CTRL i (right middle finger on the i). We can cite Rules (1) & (2).

COPS! CTRL z, CTRL 1

Properly speaking, z is little finger key, but I would imagine most people who use the combination CTRL z put their forefinger on z. CTRL 1 is no better: 1 is also a little finger key but, again, I bet most people use their forefinger here. CTRL z seems a little more fluid.

SUMMARY

when in Wordwrap, use CTRL keys at all times for all keyboard commands which have FCTN key alternatives. Do not use the top row keys at all except for LAST/NEXT PARA (CTRL 6/4) and non-Wordwrap REFORMAT (FCTN 2).

Please share your ideas if you think of good cases to justify the alternative keystrokes I choose not to recommend. The arguments presented here do not cover every case and some good, points may be derived from further analysis.

HUG LIBRARY CATALOG ADDENDUM June 1988

1320 CHARACTER & SPRITE SMAPE MAKER == 18

A graphic designing program from Gordon Leonard of Las Vegas, NV. Has many features. Comes with complete documentation. 72 sectors

1321 DUMBO**NYARC GENEVE & MYART

A MYART picture for the Myarc 9640 Geneve. It can be loaded with GIF 2. The resolution is 256 x 212 pixels. The setting for GIF 2 is MO (zero). Created and submitted by Cynthia Becker. 31 sectors

2059 TI LEDGER**IB Printer optional

This is an update to the original "Fairware" program by Don Scott. Updates by Bill Gaskill. An excellent automated accounting system. 161 sectors

4056 SURVEY**DV/80 Printer rqd.

Three text files with details on the results of a 1987 survey by Ali Ulgen from Ohio on the TI 99/4A. The comments are well worth reading. 186 sectors

4057 TI NEWSLETTER INDEX**DV/00

An index of many of the entries from the TI 99 User Group's newsletters. There are 429 entries on many different subjects. REQUIRES DS/SD OR DD/SS DISK. 575 sectors

5276 POP DENO**<u>P1/3</u>

A music disk from West Germany with 4 pop tunes including Charleston, In The Mood, Amorado, and Piohwalzer. Loads as DSK1.POP. Remainder of disk will autoload. 223 sectors

5277 HARD LUCK WORLD ** IB

A great version of the KISS' song by the same name from Gregory Rashall. 37 sectors

5278 GRESORY RASHALL'S MUSIC DISK**IB

A fine collection of songs from Gregory Rashall. Excellent job of programming. 235 sectors

9030 PHONE HAKE 88**IB

Two new phone make programs from Tom Auleta to be used with "Mass Transfer" program. Comes with complete docs. These programs fix the original PCP file editor and also include MacroMaker files. 104 sectors

9031 PAST-TERM V 1.16/2JPH**XB

The latest and possibly the last version of Paul Charlton's "Fairware" program. Revisions made by J. Peter Hoddie. Has many new features. Comes with complete documentation. 211 sectors

9032 PC-PURSUIT ** DV/80 Printer rad.

Three text files of information concerning PC-Pursuit. 168 sectors

9033 UMPACKED**YR

Another fine program by Barry Traver that is a companion to Archiver 2. This program simplifies the unpacking process for those with 1 disk drive. 25 sectors

9034 UMPACKER 2**R3/3

A "Fairware" program by Barry Boone that will unpack "ARCHIVED" files very quickly. Works with multiple drives.

9035 ARCHIVER II V 2,4°*XB

This is the latest version of Archiver from Barry Boone. Has the ability for file management including copying, deleting, etc. So far this is the best version yet. 65 sectors

TI-FAIRE

Last month's meeting, coming the day after the Dallas FLUG TI-Faire, brought an informal report from some of our members, in particular, Jim Uzzell, whose report follows:

Noted as present were the Millers Graphics Video Text card, MICROpendium, Amarillo Bytes with LOTS of "re-cycled" equipment including some unique items (a TI Double Density Disk Controller!?), and Los Angeles contingent with lots of hardware and software for sale, and even our own Kenny Espiau. The Mechatronic Mouse was mentioned (but I failed to note the "provider"). Also present were the TI-99/2 and some CC-40's, and a TI 99/8 and, strangely enough, a TI-99/5 (!).

The Gram Kracker and the Gramulator both appeared, in some form or another, not described; and Bud Mills of the Horizon Ramdisk was there, with a disk containing the improved (Loads Quick!) OS for the Geneve. Mills held the line on the prices he had advertised right through the Faire, and had plenty of buyers for the Horizon, so hats off to a TI product supplier who DID what he said he would do, instead of backing down on his word. Prices are UP ever since then, though! The Geneve New Expansion Box was either shown or described also. And John Owens and Phil Van N. of the JUG Group were there with the United Data Base project.

Distinguished by their non-appearance was Myarc's Lou Phillips (lets hope he is home working on a disk controller card de-bugging project); and Databotics Grand Ram, in a working form. Oh, it was there, reportedly, just didn't work. A perfect example of a situation where NOBODY wins—the supplier gets caught in a squeeze as prices go up and development time and costs extend, while possible sales disappear. And the would-be users do without or seek alternative devices.

By all accounts, the Faire was a success, and we hope that it was as worth-while for the FLUG and other Dallas area groups as it was for those who attended. No one came home disappointed, apparently.

Now for a complaint from my point of view, as regarde II newsletters. Seems that there are a few groups Newsletters which have a problem giving credit for articles or information which they re-print. It is fairly obvious in some cases, since the same info or exact layout may have appeared in one or several other newsletters just prior to that. It becomes a little obnoxious, though, when something is re-edited and printed or re-typed, and then later shows up in "The X,Y,Z TI NEWSLETTER" with the credits for both Original source and the Re-printing source carefully removed, but other-wise exactly zeroxed from the re-printing newsletters pages. Folks, we know you didn't type that article, because sometimes we know who DID! Actually, it is a dis-service to to the Original Source AND to Your Readers since they cannot trace back and get in touch with the original source in order to get more info OR offer updating or correcting facts back to that source. I have a nice reprint of a II console ad here I would love to run, but don't know which NL it came from, so I will hand on to it until I find out.

==Who had the ad from Interface Age?==

Corrections, additions to the March 1988 article on printers command codes...:

Elite fant on the MX-80 = 27;77 End Elite on MX-80/FX-80 = 27:80

Over-ride "Paper out".

for all EXCEPT Okidata = 27;56
Re-enable "Paper out" = 27;57

Enable Proportional Line

Spacing, S6-10, FX-80 = 27;112 Enable Prop. Line, KX-p1091= 27;111

RESET Printer (to "power-

up" cond.);56-10;10X;FX-80 = 27;64 (24 on OKIDATA)

Slashed zero; SF-10 = 27;92;1

Most of these came from another chart, from Bits and Bytes, June. 1987. A couple from other sources, and a couple I had but left out of the list. Any Others that you know of??

FUNNELWEB 4.0

IF you have Funnelweb 4.0, you know it has a few capabilities beyond TI-Writer; here are some of them:

=> In Editor, to change text from lower case to upper case, place the cursor at the start of the text and hold down CTL and; together. Magic! To change from upper to lower case, use CTL and . together. Great time saver if you want to change a group of words for headings, etc!

=> In SD (Show Directory), the asterisk between the SIZE and the TYPE columns indicates a FRACTURED FILE; one which is stored "split-up" on the disk.

Also, you may select a File from SD by pressing that NUMBER from the list, and it will become the Default File name for Load File and Save File, or for Delete File or View File, EXCEPT that it will not accept the number for a PROGRAM format file. Keeps you from trying to Load an X-Basic program into the text editor!

=> SD now Prints the directory if you press "P". => On the SD directory in Funnelweb, PROGRAM mode files can be flagged by pressing " = " and will display (in the REC column) either a BX (for a BASIC/X-Basic RUNable program), or by EA (for a Memory Image (E/A 5 type)) program to be run from Funnelweb loader #3, or remain \a blank if it is a Data file in "Program" format: "CHARA1", for instance.

=> The Editor/Assembler Loaders in FW 4.0 are accessed from the Ed/Asmblr option of the 1st Menu if XBasic is being used.

=> If you use the Funnelweb Loader #4 on a DF/80 file, it will automatically reveal the REF/DEF table names, one of which will be the "Program Name" to "Start" the program, if the program does not auto-start. It will then allow you to select which one you want to try to use. Usually only one or a few will show up, but this can be a real ice-breaker with that game your buddy gave you last year but forgot to give you the "Program Name" for (and the Name was "NEVER").

=> CONFIG program on FW 4.0 will load and run ONLY via the Menu selection CONFIGURE on the first menu, and only if you have loaded FW via X-Basic. You can configure the Editor and Formatter print devices, plus some other choices, plus putting your own programs on the menus, IF you know which of the loader type numbers matches each of your programs.

-> You may change the Print Device name AND set printer codes for condensed, etc, for DM-1000, by using FCTN - 3 at the first screen in DM1000, and then answering Y for the "Save to disk?" question. All the CONFIGurations require the write tab to be off the FWb disk, of course, so do it on the BACKUP or WORK copy, not on the original.

BACK TO BASICS: A LITTLE HISTORY

After the two articles in the last two months on types of TI equipment, cards, upgrades, etc which are available, I thought this month I would mention some of the operational aspects of the TI 97/4A system(s), and some perspective on how we got where we are.

First of all, the most basic TI system is just the console, with no module. Console Basic loads automatically from ROM & GROM, and you have a nice video of the Basic program. You didn't even have to build any of the equipment in order to get this, complete with video display, and have 14536 bytes of program space available. Big deal? Well....

THINK about 1975 when the Altair was advertised in Popular Electronics, without a Basic, and with (?) 512 bytes of memory, if MY memory serves me. The original Altairs were in two boxes, each about the size of the PEB box, after you built the kit, and had no keyboards. It was around \$500.00 or so, as I remember, and that was when engineering grads got salaries of \$1200/month, maybe. It pretty quickly became a couple of k-bytes, and then "Apple" got its start from a 4-k expansion board for the Altair. This board used 256 x 1 chips, about 32 of them per 1 k-byte, or maybe it was 1k x 1 bit chips, then only 8 of them per k-byte would have been needed. Big board spaces needed, and lots of patience, since there was just 8 "bit data" switches for inputting each byte, and no video display either, at first. Hours and hours of fun just to add three numbers together! But it WAS programable!

The first HDME computers, meaning usable by the average non-hacker, non-computer/electronics-professional, were the Atari 400, Commodore VIC 20, TI-99/4, Sinclair Timex 1000, Apple II (was there ever an Apple I ?) Radio Shack TRS-80, and of course you remember the Panasonic JR-200, the Mattel Aguarius (4k), the NEC PC-6000, and the Spectavideo 318 (32k), don't you? These all came about from about 1978 thru 1982, and all had from 2k to 16k of memory, with expansion modules in some cases, all had a BASIC programing language for the console and displayed the program on a TV screen. Most had programs available on cartridges for games and sometimes for home and office applications, and all were able to Load programs from cassette tapes, and, most important, all were able to SAVE Basic programs to cassettes. Even the Original IBM-PC was only (I think!) 16k, and could save to cassette. The market at that time was mostly persons who were "computer curious" and who believed that the educational aspects, fun (games), and home/office (applications) uses were worth the costs. They soon found that purchasing computer consoles was just the start to "Computer addiction".

People working with computers or terminals at work and in schools knew already that they wanted more than a console with a cassette for storage—they envisioned printers, disk storage, and other fun things that could "do more, faster". Hanufacturers were more than willing to churn out new models and new complete lines of computers, with lots of (expensive?) peripheral devices, and the flood was on.

Many lines and brands of computers for both Homes and for Offices/Schools have been superseded and abandoned, gone the way of the Curtiss aircraft, Studebaker car and the Osborne computer, not so much as a result of bad design or poor understanding of the Market and the product needed, but as a result of the rapid change in the market-place ON THE SUPPLY SIDC. It was not only the purchasing "public" which was un-knowledgeable, but also the manufacturers, who found that their facts and expectations did not match the rapidly changing nature of the computer marketplace.

The suppliers to these markets contributed to the overall confusion by continually introducing new systems and equipment which was often incompatible with the previous line, compounding this error with an over-emphasis on their Hardware and "cimmicks" rather than on the Usability and Usefulness of the systems. Few manufacturers made a genuine effort to encourage software development for their machines, outside of what they could make money on directly, and the success rate of many machines was exactly the inverse of the manufacturers "Software Greed" level, which attitude itself was probably a holdover from the "Games Machines" marketing attitude toward software. It should be noted that the 99/4 was on the market from mid-1979, and had the largest amount of manufacturers-generated software of all these computers by mid-1983. Apple apparently did not attempt to control the software supply for their machines, however, and the Apple II+ had the largest software library, via third-party sources. Apple made its millions by selling (relatively expensive) HARDWARE to the people with the most money to spend--the school districts. The success of the Apple II+/e/c and the Commodore 64 was highly dependent on the large amount of usable, useful, affordable software and hardware available (not limited to the manufacturer!), AND on the continuity of and support for the product line.

Manufacturers with GOOD machines which had excellent design and features, but which cost a lot to make, found that they were caught in a price squeeze in which newer designs could DO the same things better, and be made mure cheaply. Yet if they continually re-designed and started new (and non-compatible) lines, this caused them to lose the software base in the market, and led to even greater "new line" costs to re-write all the old software. And so the Osborne, PC-jr, and 99/4A, and others, developed the unfortunate condition of "bottom-line-itis abandonitis", typified by heavy marketing expenditures and corporate verbalizations of optimism and continued support, just prior to pulling the rug out from under the uses of their (previously marvelous) hardware. The real gentlemen of the marketplace have stood behind the equipment, and certainly Texas Instruments has done well in this aspect. What other company still supports the users of a piece of equipment they quit selling 5 years ago, after losing millions on it? Complain if you will about the way 'TI handled the 99/4A and what they "should-of" done back them; they have done a good job of support for us since then.

So our II-addict decides to add some (expensive) games and "mducational" modules. Now he can really 80 something, or some things. But now he needs to save his "Done" results, so he springs for a II cassette and cable. Great, we now have "Home Programable" system with an easy way to input a program and data (keyboard and Basic) and easy viewing of the results (screen), and it can permanently SAVE and recall both the Program and the Data. It took from 1938 to 1978 to get this available at a "Home Affordable" price. And it is easy to use. SAVEing and re-loading (OLDing?) programs are as easy as the commands: no file names or complicated code numbers to remember, just a location on a tape. Somewhat limited though, since only Basic programs but not Machine Language programs could be SAVEd. But, DATA files could be created and saved to and from cassette. Did you ever take advantage of this ability? Neither did I! Somehow it seemed a lot easier to use variable names and arrays than to figure out all the circular logic on file types and manipulation from pages II-118 to II-135 of the infamous 'Green and Yellow' book. In fact, it is actually possible to create and save M/L files to and from cassette, with the Mini-memory module. And supposedly M/L files can be loaded with Extended Basic via the CALL INIT; CALL LOAD("CS1"); CALL LINK("name") series.

Now our TI user sees other people using fancy disk drives in their systems, and learns (not from the TI manuals, either), that something called a P.E.B. box is necessary for him to do this. He finds himself buying one of these, and controller card and disk drive and memory expansion card and even something called "Extended Basic" which will allow him to use all this stuff effectively. And since his wife is bugging him about what all this expense will really DD, he needs to buy a printer and an RS232 card to run it. Plus, of course, TI-Writer to write the letters to show his wife, and a resume' since he now needs a better job to pay for all this addiction. So, with our revised definition of a "basic TI system" as listed above, we are ready for some serious computing.

LOADING PROGRAMS from CASSETTE

First of all, I will say that I DON'T Know All About the machine operations of the 99/4A, and that is one reason I have been digging up information of this type. Now I have an excuse for trying to put it into some coherent form.

I. Console Basic only accesses (14536 bytes of) the VDP memory, even when operating with PEB, and therefore can only load a Program to that area. In fact, with disk, it actually only has 12448 bytes available, unless CALL FILES(1) is used, then 13480 bytes are available for THE PROGRAM PLUS ALL VARIABLES AND VALUES created during RUNning. And each variable, when assigned a value, occupies (9 bytes + name length) more memory than before the program started to RUN. The disk controller itself allocates 2088 bytes out of VDP--you can regain 1036 bytes of this by:

CALL FILES(1) (enter)

NEW (enter)

OLD CS1...etc

- 2. Extended Basic itself uses 608 bytes more memory than Console Basic, leaving 13928 bytes without a disk or only 11840 bytes for Program storage and RUNning space with a disk, or 12876 bytes if CALL FILES(1) is used. HOWEVER....
 - A. Programs use more space RUNning than just "as typed".
- B. The computer needs a 256 byte buffer to SAVE or OLD with a cassette, so maximum Program size to be "cassetted" is actually (approx.) 14280 bytes in Console Basic, or 13224 bytes if disk system is present and CF(1) has been done; and 13672 bytes with X-Basic, no disk, or 12620 bytes with X-Basic and a CALL FILES(1).

So our TI'er will just fix all this by putting in the 32k memory expansion, then when we do a "SIZE" we see that we have 24488 bytes of PROGRAM space and 11840 bytes of STACK available. Great, now we can load up from cassette! Wrong! What happens here is strange, but the cassette uses YDP memory as a temporary buffer before transferring the Program into or out of the 24k Memory Expansion, and ONLY 12620 bytes can be handled, IF a CALL FILES(I) has been executed. This is 11840 - 256 + 518 + 518 bytes. And this equals 49 sectors on a disk, which will cause the program to be in Internal/Variable 254 form on the disk, which to some extent bypasses the CALL FILES activity, but makes reversing the Loading procedure to a new SAVE CS1 tricky. It is possible to get a program OFF of cassette using the disk and X-Basic and then NOT be able to ever put the program BACK onto cassette! Also, if CF(1) is used Defore SAVEing a long program to disk, if must be used every Time before OLDing the program back...a pain! Remember, though, that for disk or cassette, BASIC can OLD or SAVE an X-Basic Program. So, to get a longer X-Basic or BASIC file from cassette TO disk, try BASIC and use the CF(1) trick, but use X-Basic to load it and RUN it if it is an X-Basic Program. It IS possible to play tricks on the "VDP Top Of Memory Address" word at -31808, A, B but I don't recommend it.

The most RELIABLE method of getting a Program off of a cassette and onto disk, if CALL FILES(1) and BASIC doesn't do it, is to use X-Basic and MERGE:

- A. Using Basic and the PEB off, load the program from cassette. Type LIST 32767 (enter) to LIST the last line of the Program, and write down this line 4.
- B. Pick a line # about 20% into the Program, and use Function-3 to grase Lines from the start down to that Line! Then SAVE it back to cassette.
- C. OLD CS1 the original program again, and erase about an equal number of lines AFTER that reference Line #. Again, SAVE the shortened Program back to cassette.
- D. NOW turn the console off, PEB on, X-Basic into console and turn it on. Bring up X-Basic and Load (OLD) the first of the "shortened" programs from CS1, then SAVE it to disk as SAVE DSK1.SHORTA, MERGE .
 - E. Then do the same for the second short Program, as SAVE DSK1.SHORTB, MERGE
 - F. THEN type:

NEW (enter)

MERGE DSK1.SHORTA (enter) and wait a long time.

then: MERGE DSK1.SHORTB (enter) and wait a long time.
then: SAVE DSK1.name (enter) and you should have the
complete program on disk in the proper form, and loadable as

complete program on disk in the proper form, and loadable as DLD DSK1.name. Remember to RUN it in X-Basic or Basic as required.

PROGRAMS, FILES, AND LOADING

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 Tigercub TIPs, plus a reprint of Jerry Bentzinger of the Omaha TI UG, May 1986; Jim Hakeel, Northcoast 99ers; Rich Klein of the Chicago Tlmes; 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. SD....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.
 - Data Files--those with multiple occurences 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 Program(s). If 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.

8. 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.

C. 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 DV-80 files are TEXT files and sometimes contain the documentation or instructions, in which case they may be named "XYI/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:

Rec P Filename Size Type

Filename==This indicates the NAME of the File: it will be refered 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 orocedure.

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

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

"1/V 254" Internal/Variable "I/F 128" Internal/Fixed

"PROG" Program (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.

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 "X-Basic Games", you will find mostly "Program" type Files and possibly some "Int/Var Z54" type. You may find ONE File named "LOAD", and possibly one or several "Dis/Fix 80". There may me 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 X-Basic. If the Program is made to run WITH the X-Basic module, it will almost always have a "LOAD" file and multiple files of "PROGRAM" or D/F BO types which the "LOAD" File accesses and Loads.
- 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", "GAMG". etc.

Other disks may have a series of "Program" format Files which will not Load into X-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 mames such as "UTILI", or a series of sequential names. cont...=

- 4. A disk full of text files will USUALLY be "Dis/Var BO" 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 RLE 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.
- 6. Some other file types such as Internal/Variable 128 may by 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.
- ALL types of Files may show up being used for DATA Files, depending on the Disk and Program involved. Int/Var 44 and Int/Var 128 seem to be common in this form. Any Int/Var 254 Files of LESS THAN 45 sectors will NOT be X-Basic Programs, but rather Data or Text Files.

PROGRAMS

- 1. The easiest programs to Load are those which are in module form, in hich 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,
- 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 them auto-loads the portion of the disk which is being used. The documentation may be on the disk but usually is in a users quide and ranges from very skimpy ("hints") to downright over-whelming (the manual for TI-Writer, for instance).
- Cassette was the first storage medium 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 "GLD CS1" for both BASIC and X-Basic, as per the Users Reference Guide (green book) op II-40 to II-42. See also the previous article on TI-Basics for Loading information.
- 4. A. RUNable Programs (for Basic/X-Basic) on disks are usually "PROGRAM" format up to about 46 sectors, or, for XBasic Programs longer than 46 Sectors, "Int/Var 254". These are loaded and run via "OLD DSKx.filename", and then "RUN", or for X-Basic, via "RUN "DSKx.filename" which loads and starts RUNning immediately. Maximum size of Int/Var-254 X-Basic Files is 97 sectors, and they only exist in Memory Expansion systems.

- B. Following are TIps to determine Basic/X-Basic type, since Basic/X-Basic will each Load the other type of Programs from cassette or disk, but not necessarily RUM properly:
- Most BASIC Programs will load and RUN in X-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. X-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 X-Basic. IF it LISTs ok in BASIC and has ALL single-step LINES (no double colons between statements in a LINE) and ND statements such as "DISPLAY AT"; "LINPUT"; or anything whatever to do with SPRITES: then it probably is in BASIC.
- C. Another (sub-routine or partial Program) form used with XBasic Programs is the "MERGE" format: Dis/Var 163. These can be loaded via: HERGE "DSKx.filename", to combine with the Program already in memory, and then RUN. Any regular Program which is saved via SAVE DSKx.filename, MERGE will be stored in this MERGEable Format--Dis/Var 163, and must be re-loaded via the MERGE command.
- D. ANY XBasic PROGRAM File with mame "LOAD" will auto-load and RUN if it is located on DSK1, when the X-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 34 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
- PROGRAMS, but rather some form of specialty 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.
- 6. Generally, for PROGRAM files (and Int/Var 254 files) which appear by CONTEXT of the disk or FILENAME to be RUNable Programs files, try X-Basic "OLD..", then if they will not load, try using E/A 5 to Load them. If they STILL will not Load, they may be DATA files.
- -----Funnelweb SHOW DIRECTORY will indicate whether a "Program" File is XB or EA type, also: if you press " = '" while on the directory screen, FW 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.

cont...=>

- 5. A. MEMORY-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/X-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 Funnelmeb Loader 3 (in Ed/Asable Menu Screen, Loaders Option). These Files will give an "I/O ERROR 50" if you attempt to Load them via XBasic. 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:
 - 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, when the Last File has been Loaded.
- C. LOADING a Memory Image File series can be done from Ed/Assembler Option 5, by entering the device and name of the FIRST File of the series:

DSKx.GAME1 (enter)

and the Loader will toad this File, search for and toad 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-Mriter Option 3, "Utility" and Funnelweb Loader 3 operate in the same way. It might be noted here that the TI-Mriter 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 DM1000 files MGR1 and MGR2 to FORMA1 and FORMA2, you can Load DM1000 by choosing the "Formatter" (Option 2) on the TI-Mriter module, as an example.

JUNE 88 - Houston User's Group - Page 11

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 to:
 - 1. Chopse E/A #3 or FN (Loader Menu) #4 or M/M #1;
 - 2. At the prompt "FILENAME?", enter the Source and Filename(s): (there is no Default Name built in) DSK1.FILE1 (enter)
 - Loader will prompt for "FILENAME?" again: OSK1.FILE2 (enter) (assuming a Filenames SERIES OSK1.FILE3 (enter) are present on the 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", "60", "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("DSKx.filename")
CALL LINK("program name")

except: Compressed Object Code cannot be Loaded through
X-Basic. cont. =>

NOTE: X-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, X-Basic will not resolve external REFerences between Files and requires EQUates into the GROM 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 GROM. 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 X-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 leage file Loaders, or which are HYBRIDs and contain Machine Language sections "hidden" within what the 99/4A considers to be the X-Basic Program, either directly or by CALL LOADs from DATA statements. The File "LOAD" in Funnelweb, which Auto-RUNs in X-Basic, is one such "Hybrid" Program, with over 25 sectors of Machine Language code. Obviously, attempts at EDITing such an X-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.

▲ Little Louder Please •

ers From West Penn 99



F. I hope that all of this does not leave you more confused that ever. -I- certainly had a time trying to sort out and piece together all the little bits of information relative to the Types and Methods of files and loading used on the 99/4A. If this is of any help to you, or if you spot something I missed or just simply messed up on, please write and let me know. Good luck, and happy RUNing. r.lumpkin. ______tl May 88--Desics I-B



JERRY D. ILLING OWNER 2530 W. HOLCOMBE BLVD. HOUSTON, TX 77030 (713) 666-9880

1 1 10% discount to HUG members 1 1

HUG Newsletter

c/o Richard Lumpkin 10910 High Knob Drive Houston Texas 77065

BULK RATE : ; U.S. Postage ; PAID : Bellaire, Texas : ! Permit No. 644 !

Membership Renewal Status

(four digit code left of your name)

If code contains one of the following:

"+" : membership expires with this issue

"**" : dues one month delinquent

"***" : dues two months delinquent

"LAST ISSUE" : Membership deleted!!!!

8803 Dallas TI Home Cmptr Grp P.O. Box 29863

Dallas, TX 75229