FUNNELWEB Vn 4.40 Oct /30/ 91
Universal Utility Environment

(1) General Notes

FUNNELWEB Vn 4.40 is a powerful operating and utility system for the TI-99/4a computer. It will run on a minimal disk-based system with 32K expansion as an Extended Basic program (TI or Myarc XBII), as an assembly program file (E/A or TI-Wr), or with a Horizon style auto-booting RAMdisk with any or no cartridge at all. It supports in a single seamless system a complete and improved range of standard utilities (E/A, TI-Wr, Disk Manager) and menu/loading facilities for a wide range of other programs. Cartridge swapping from XB is now largely unnecessary. The basic TI-99/4a system was always and still is the easiest of computers to use, and FUNNELWEB makes it just as easy for the advanced facilities too. The more powerful the system the better it goes. A further set of files is available to exploit the extra power of V9938 based systems, the DIJIT AVPC card, the Mechatronics 80 column unit, and the Geneve 9640 in GPL mode. Recent versions give some support for hard disk systems using the Myarc HFDC so that the FUNNELWEB system can be run from its own directory on the hard disk, leaving the "DSK1 emulation" free for use as work-disk.

The program has been written entirely at Funnelweb Farm and is distributed as "FAIRWARE". It is not to be sold nor distributed with excessive copy fees, nor ADVERTISED as part of ANY commercial sale, nor placed on copy-protected disks. Placing of these files on any electronic network or BBS without explicit permission (to be renewed for each new version) of the authors is expressly forbidden.

The Vn 3.0 and later programs are NOT IN THE PUBLIC DOMAIN, but "fairware", with all rights reserved by the authors. No responsibility is accepted for consequences of its use. Please refer to the Fairware Notes at the end of this file. The FUNNELWEB package is issued with no commercial copyrighted utility programs on it, and we request that the package be passed on in its original form only, complete with ALL document files.

The equipment that you will need to run FUNNELWEB is the same as needed for TI-Writer or E/A except of course that the specific module is not necessary.

Absolutely necessary

(1) TI-99/4a console
(2) 32K memory expansion
(3) Disk drive + controller
(4) Module/RAMdisk to load it

Highly desirable

(4) Two or more disk drives
(5) RS232/PIO and printer
(6) A RAMdisk or several
(7) V9938 80 column system

Other items that you will need to make full use of FUNNELWEB Vn 4.40 are

(1) E/A, TI-Wr manuals
(2) c99-REL4 package
(3) Your utility files

(2) Files and Documentation

~~~ ~~~~~~~~~~~~~~~~~~~~~~~

The FUNNELWEB package contains a number of files besides this one (50 files in all). These may be wholly or partially archived on the distribution disk(s), using Barry Boone's Archiver III program - also "fairware",

LOAD, FW - the main program in XB and E/A program formats

ED,EE - files for both Text Editor and Programmer's Editor

FO,FP - Formatter program files

AS,AT - Assembler program files

CF,CG - Configuration program to set preferences in FW and LOAD and to set up User Lists

SYSCON - Sample configuration file for use with CF/CG

C1,C2 - character definition files

QD and QF - Quick Disk Directory called up by <AID> from main program or by the Assembler and Formatter

DR,DS (or perhaps on disk as DR40,DR41) - DiskReview program called from Central Menu Option #8 or as initial autoload

EA,LL,SL - system auxiliary loader files for EA Program and Load/Run, LowLoader, and ScriptLoader

UL,D1 - sample User List files

ML - a sample Multi-List file

SCRIPT - sample load script

LH - LineHunter assembly programmers' search utility

FSAVE - improved Save utility for generating E/A program files

CP,C99PFIO;O - auxiliary files for use with Clint Pulley's c99
~~~

Rel 4

CT8K/O - installs FUNNELWEB as menu item in E/A + 8K RAM cartridges

LDFW - Dis/Fix 80 loader for FW from Minimem, E/A and Myarc XBII

XB4THLD - loader for TI-Forth from XB cartridge.

Detailed documentation of the various aspects of the program is to be found in the FWDOC files,

FWDOC/LOAD - general information on system and disk organisation, the XB user's list, and program-wide facilities

FWDOC/TIWR - use of the package as a substitute for the TI-Writer module

FWDOC/EASM - programmer's editor and assembler operation, and program loading functions

FWDOC/UTIL - notes on utility programs in the package used with FUNNELWEB

FWDOC/REPT - chronicle of bugs, fixes, updates, problems, and background information. Make sure you read this file

FWDOC/SCLL - details of Low-Loader, Script-Loader, Assembly MAKE, and User and Multi-List files

FWDOC/DR40 and FWDOC/DR41 - use of the DiskReview menu option in the 40 column version.

The supplementary file collection for 80 column use with the DIJIT AVPC and other V9938 based systems contains

ED,EE (may be found as ED80, ED81 on the distribution disk) - 80 column Editor files which also have a 40 column edit mode

DR,DS (may be as DR80, DR81) - 80 column DiskReview program

ML (may be as ML80) - the 80-column Multi List program

FWDOC/EDAV - supplementary editor and general docs for the 40 and 80 column operation with AVPC, TIM, Geneve, Mechatronics systems

FWDOC/DR80, /DR81, and /DR82 - docs for 80 col DiskReview

FWDOC/PSRV - details of useful program services, pointers, and data available in the FW interface block. See also FWDOC/REPT.

In order to fit the files on a DSDD disk they have been partially archived. Enough files (-READ-ME, LOAD, FW, ED, EE, DR, DS, QD) are left unarchived to get started on any system. Barry Boone's

FAIRWARE Archiver can be used to unpack the other files. The complete unarchived set of files will normally be supplied only on an 80-track DSQD disk. The DOC files in the 40-col group contain Formatter commands, while the DOC files in the 80-col group should be printed from the Editor. In particular FWDOC/PSRV contains assembly source code examples containing "at" signs.

(3) Update Notes
~~~ ~~~~~~~~~~~~

Vn 4.40 is a further refinement and extension of the FUNNELWEB system, and retains full external compatibility with prior versions. Significant changes, apart from minor bug-fixes, from Vn 4.31 are

(i) A flashing cursor with auto accelerating repeat has replaced the static sprite underline cursor, with timing delays compensated for processor speed. GROM address setting now should support Module Library devices. More externally accessible program services and new documentation are provided.

(ii) DM-1000 files MG/MH and Disk-Patch file DP were dropped after Vn 4.30. They still can be used with Vn 4.40. Use of the LOAD only reload path from DM-1000 is no longer advisable.

(iii) Script-Load and the Assembler now support a multiple file MAKE function.

(iv) Error indication for Duplicate DEFs and Unresolved REFs has been improved in the object loaders, and Script-Load now has a full error handler with extended Unresolved REF display.

(v) The 80 column Editor now supports a 40 column Edit mode. Program file checking has been dropped for extra speed from the Editor SDs as superfluous now DR is here, and the original TI RE bug is fixed. This has made room for --

(vi) Double-View in the 80-col Editor which allows page scrolled access to parts of one or more files from SD without further disk activity.

(vii) DiskReview now has some support for DSKU file comments, and file read in for View is faster, and some error reporting improved.

(viii) In UL files the <esc> path has been modified to suit better the revised <esc> handling in the main program. It would be a good idea to transfer your existing lists on to the new template by Fetching the old file, Making Reserve of it, Fetching the new UL, eXchanging data, and then Saving back under the original filename.

(ix) A new class of Multi User List files has been introduced.

(x) Various other auxiliary system programs have been revised, in particular CF/CG, LDFW, CT8K/O, LL, SL, AS/AT, and ED/EE.
~~~

Replace all files to be on the safe side.

(4) Fairware Notes
~~~ ~~~~~~~~~~~~~~

Your letters and contributions in appreciation of this program will be welcome. Many suggestions from correspondents have found their way into the system already. If you wish to interface to FUNNELWEB at the assembly language level, consult the FWDOC/PSRV and FWDOC/REPT files and articles (Living with Spiders) that have been published (not much updating needed for this version) and contact us for further details if necessary.

The fundamental design policy remains that all capabilities be accessible with the XB module in place, and to create a system which makes the most of the strengths of the TI-99 system, without straining to imitate any other OS. The FUNNELWEB system is naturally designed to enhance the hardware that we have here and developed on that hardware. Extensions to exploit fully new hardware really require hands-on development, but even so wherever possible system extensions are made for significant new or other hardware that comes to our notice, using whatever information is available on device capabilities and foibles, and just flying blind. The system in use has 2 Horizon 192K RAMdisks, Quest (HV99) 512K RAMdisk, TI RS232 card, Myarc FDC, 2 Chinon DSDD and 2 National 5" 80 track drives, various E/A + RAM cartridges, and the DIJIT AVPC card. The backup system has Myarc 512K RAMdisk and Mechatronics 80-col unit. Specific extra program capabilities or bug work-arounds have been made, in the flying blind category, for the Geneve 9640, and Corcomp FDC. Experience with the Myarc HFDC has been so bad that plans to support this device in DiskReview have been abandoned.

No tangible assistance or direct cooperation for the FUNNELWEB development has come from ANY hardware maker, with the exception of DIJIT Systems, whose AVPC card (sadly, no longer produced) gave a major new direction to the programming effort and reason for staying with the TI-99/4q. Also Myarc Inc returned the XB-II files to us on our own disk with no other form of communication ever, following repeated queries on XB-II and advice to them of disk DSR bugs. Gary Bowser of OPA provided the Vn 8.14 HRD ROS. All support other than these items has come from individuals and User Groups.

Any commercial software or hardware producer wishing to claim compatibility of a product with Funnelweb in advertising is advised to have us check this first.

As for each "fairware" user's obligations, we can suggest only that you judge the program on its intrinsic merits, best measuring its worth by how much you use it as compared to other "fairware" or commercial programs that you have. Even individual components of the package are as substantial programs as many that are offered as commercial items at $10-20 or more apiece. If you use it, be honest enough to take the trouble to show your appreciation, but if you try it and decide not to use it, then please just extend the free trial by passing it on to someone else. Our normal practice is to send
~~~

the very latest revision in response to substantial contributions, so please indicate the latest version you have, and your disk capabilities when you write to us. FUNNELWEB is NOT sold to order, and any $$ received are assumed to be in appreciation for use of past issues.

If contributions are made by check (Postal IMOs are no longer accepted by Australia Post) they should be made payable to Tony or Will, or Will alone (rather than "and").

All letters needing an answer will be responded to sooner or later, but we just can't afford the time for, or cost of routine mailing of updates and don't volunteer to do so. If and when updates are issued they will be placed in distribution through our regular, and/or recent contacts, and we do not offer to provide distribution from Funnelweb Farm at near nominal cost or less. Any renumbered issue represents a great deal of work, not always immediately apparent and which gets more and more difficult as the limits of the 99/4a are stretched. Our norma practice is to release minor fixes and improvements as soon as they are made. If you write new utilities for FUNNELWEB please to let us know the details.

Oct 30th / 1991
Tony and Will McGOVERN
215 Grinsell St.,
Kotara, NSW 2289
AUSTRALIA

FUNNELWEB Vn 4.40 DISKREVIEW

(1) INTRODUCTION

DISKREVIEW (program files DR/DS or as DR40/DR41) integrates with the FUNNELWEB Vn 4.40 system an extensive set of disk snd file utilities. It features an enhanced Directory function like that in the FUNNELWEB Editors. In addition it provides all file marking functions of the FUNNELWEB QD, most disk manager functions needed in normal usage, a sector editor with string search over individual files, an alternative interface to the FUNNELWEB main menu, and complete access to the FUNNELWEB file loader system from the cursor bar in the directory display. In addition an Extended Basic program loader handles both Program and I/V254 formats. The View function supports full bi-directional scrolling within a circular buffer of 8 Kbyte size, and printing out from the buffer of the Viewed file. The program is fully compatible with 80-track disks using the Myarc FDC with appropriate DSR ROM, or Myarc HFDC (but this device appears to have serious bugs in its 80-track file copy DSR routines). Every effort has been made to keep the program insensitive to the various incompatibilities of the Geneve 9640 with the 99/4a. This file is continued in FWDOC/DR41.

(2) LOADING

It is installed permanently as Option #8 on the FUNNELWEB E/A Central Menu screen as DISKREVIEW, filename DR, with an Option 1 (Text-Mode TI-Wr) load path forcing C1 load to refresh the TI-Writer control character patterns. DR can be loaded as an Option #2 (GPL) file if desired from another Central Menu or User List entry. It calls on FUNNELWEB internal routines which means it will crash if you load it from E/A 5. Screen colors are set up as for entry to the Editors from the FUNNELWEB main menu. See FWDOC/LOAD for discussion of initial auto-loading options.

(3) PRINT FUNCTIONS

All print functions use as default the FUNNELWEB print device as installed or edited for use by the Editor PF function in the main program. No provision is made in the program for explicit control sequences to be sent for setting up the printer, but these are easily handled by using separate short files containing the necessary codes and printing these from View before and/or after printing other files or directories. All or marked parts of Display files of any record length may be printed, with original record lengths preserved. As the print device name can be edited to be a disk file this gives a limited file copy function for the whole or selected parts of Display files to a target device with general pathname.

(4) DIRECTORY

When DISKREVIEW first runs it shows a screen with information block, command reminders, and window for alternative menus or directory paging. A plain pop-out window in the lower center of the screen is used for user input, progress reports, and error messages. All low-level disk accesses, other than the Myarc FDC direct formatting, use standard DSR sector access or other DSR subprograms which require a unit number entered in the form DSKx., with the character "x" being in the range 1-9. If your choice can't be found then the error trapping will tell you. When q disk directory is read, program file checking is done automatically. The disk/file information block shows the usual disk and drive details. The block of filenames shows certain filenames of recognized types which may be altered by marking.

WF :- the current FUNNELWEB workfile DV/80 or DF/80 display file as would come up for Editor LF/SF etc.

OF :- the current FUNNELWEB DF/80 object filename as appears as default for the file loading options.

PF :- the current FUNNELWEB assembly program file name as appears for program filename default.

On entry to DISKREVIEW the current filenames are read and displayed, and on exit the names showing are installed as defaults for FUNNELWEB.

(5) KEY COMMAND FUNCTIONS

When DR first loads the left side key advice panel shows a number of alternatives. This screen is also reached by <enter> or <escape> from the directory screen (<escape> is either <fctn-9> or <ctrl-C> throughout DR). All key entries are automatically converted to upper case except where lower case might be needed as in ASCII search string entry.

<1-9> keys. The directory is read from the disk in the indicated drive and displayed in the right side box. After the directory has been read in, DR will flash up a "Check programs" message and continue accessing the disk to check details of any program files.

<fctn-4> (Break) terminates the process. The entries written up in the file box are not erased immediately, but are otherwise ignored by the program.

<0> key. As in FUNNELWEB this cycles the screen color

choices. Also available on other directory screens.

<D> key. This brings up a menu of disk oriented functions in the right box. See FWDOC/DR41 for more details.

<F> key. The current FUNNELWEB Central Menu entries are presented in two blocks. Selection is by reversible marker with <E/X/ /S/D> using <enter>.

<ctrl => key. Returns to FUNNELWEB from any screen where it is active,

<ctrl A> or <fctn 6> Switch to the Disk Utility screen unless a valid directory is present to be resumed.

<esc> or <E/X> keys. Returns to valid directory display if present.

A variety of key functions is available after the directory has been read into memory, and a reminder of these is given in the column down the left hand side of the screen. Some keys function only after a file has been read into the View buffer. Individual file oriented disk manager functions are executed from this directory display, and others after tagging a set of files. Delete works only on tagged files as an extra layer of caution. Entry to the sector editor is also from this screen, as many of its functions are file related.

<fctn 8> key. The usual REDO key does just that and re-reads the directory from the same drive.

<enter> key. Returns to the initial selection screen.

<ctrl C>, <fctn 9> keys. These provide the <escape> from error conditions, or to go back to an existing directory from the drive selection screen.

<E/X> and <fctn E/X> keys. These move the cursor marker up and down the directory, paging it as necessary.

<ctrl E/X> and <B/N> keys. <Ctrl X> or <N> pages the directory display towards higher numbered pages, and <B> or <ctrl E> pages back towards the start of the directory.

<space-bar> key. This checks the type of file currently marked by the cursor. The filename will appear as qppropriate in one or more of the block of filenames in the lower right block.

<O> key. <O>ldfile restores the various marked filenames to those in force when DISKREVIEW was invoked.

<T> and <ctrl T> keys. <T>ag the file under the cursor bar, to leave a visible marker in the center column. Use <ctrl- T> to tag all files, The total size of tagged files is indicated as sectors allocated at the bottom of the file-size column.

<U> and <ctrl U> keys. <U>ntag the file under cursor. Untag all files with <ctrl U>.

<ctrl A> key. Several <A>ctions on <A>ll tagged files are available from a new command set which appears in the key advice block. Details are in the next section.

<R> key. <R>unProgram sends the marked filename to the appropriate FUNNELWEB loader or to a internal XB loader. Details are discussed more fully in a later section.

<fctn R> key Allows the file under the cursor to be renamed. Edit the name as presented and <enter>. The directory will be re-read to verify the change.

<fctn C> key. Allows copying of any single file at a time from the directory drive to another drive or to the same disk under another name. The target drive number and target file name are edited and on <enter> the disk number is checked in the range [1..9]. Disk swapping in a single drive is prompted if drive number and filename of the target file are the same as for the source file. Copy uses the disk DSR direct file input and output subprograms which must be supported by the disk or RAMdisk DSR to TI specifications. Copy buffer size is 46 sectors.

<V> key. Allows viewing of ALL file types of any record length to the screen. The first page of the display is presented and it then halts for further instructions. See View/Print for more details.

<ctrl V> key. Similar effect to <V> except that the display continues in line scrolling until a further key is pressed.

<I> key. Inspect and edit sectors. See FWDOC/DR41 for details.

<P> key. <P>rintDirectory causes the current directory to be printed out to the FUNNELWEB print device (as transferred to the Editor as Editor PF device). The print device name may be edited but is used only within DISKREVIEW. The device is opened in APPEND mode.

(6) TAGGED FILE OPERATIONS

Several file operations are performed on the selection of files tagged in the previous screen.

<C> key. <C>opies all tagged files to another drive under their existing names. A list of up to eight drive numbers may be specified, and each tagged file is copied to the listed drives in turn. To stop the copy process hold down <fctn 4> BREAK which is checked after each file is completely copied. Single drive copying is not supported

under file tagging, only for single file copies with <fctn-C>. If for any reason a file is left incompletely copied, its directory entry should be deleted from the target disk.

<U/P> keys. <U>nProtect or <P>rotect on all tagged files. The directory is re-read so that the result can be verified.

<D> key. <D>eletes previously tagged unprotected files from the disk. You are asked if verification is needed for each file and if "Y" the tagged filenames are presented one by one for deletion. The directory is re-read after all tagged files are deleted or <esc> taken. A bloop will sound if the delete fails, which should remind you to check disk or file protection. If you delete the wrong file, IMMEDIATELY use the file recovery procedure from the Disk Utility menu.

<N> key. The DSKU disk manager program by the late John Birdwell allowed annotation of individual files, but in a way not recognized by DSR operations. These notes are destroyed by the normal file by file copy process. If <N>otes is pressed the tagged files are examined for DSKU type annotations and these are transferred to the files of the same names on the target disk. No provision is made for entering or editing these directly. See DSKU Notes at the end of FWDOC/DR41 for technical details and how to use the sector editor to create these.

<ctrl E/X> keys. These page the directory to allow checking of what has been tagged.

<E/X> keys. These force the display back to the normal directory commands before scrolling the cursor.

(7) PROGRAM LOADERS

The Run Program option gives access to the FUNNELWEB loading system directly from the directory display. As far as it can DISKREVIEW checks the type of program file before selecting the appropriate loader and then offers sub-choices corresponding to the various FUNNELWEB Loaders screen choices. Once the load decision is taken for the assembly load path you are back in the FUNNELWEB environment.

o Object files

Any DF/80 file presented is assumed to be an E/A object file. An object file may be a file normally loadable by FUNNELWEB, a special file that uses the Low-Loader option to load in low memory over the usual E/A utilities, or finally an auto-starting file that does not need to return to the loader. Any other DF files are ignored.

o Text Files

If a DV/80 file is selected the loader gives a warning message, and if you elect to continue treats it as a FUNNELWEB Script file for loading and linking one or more (up to 15) object files. All other DV files are ignored.

o Extended Basic programs

If the initial check of program file headers has determined that the marked file is a program with a Basic/XB compatible file header, or else an IV/254 file is selected, then DISKREVIEW checks if a TI XB module is present and uses an interrupt driven loader to start XB and load the nominated file. Because absolute GROM addresses are used this may not work with other variants of XB (not available here for testing purposes) Reports are that Triton Super-XB works fine, but most definitely not Myarc XBII. For these you will just have to do it by hand or else use the XB User List in FUNNELWEB LOAD which should work with all known XB variations, as it is invoked as a program statement from within a running XB program.

o Program files

If the program file has been identified as E/A compatible it will be passed to the FUNNELWEB Loaders with choices presented as for Options 1-3, of which 2 GPL is the normal selection, corresponding to E/A 5. See FWDOC/EASM for details. If the program file is not a recognised E/A type a warning is given. Files of this kind that are executable as assembly program files usually load into cartridge RAM, or else are extra long files prepared with FWSAVE.

(8) VIEW/PRINT FILES

The View file function presents Display or Internal files, of any fixed or variable record length, on the screen as an ASCII character display. This may not be very relevant or readable for Internal files where bytes are usually in an internal machine representation (many Display files, usually Dis/Fix 128, may also be found to contain other than the normally readable ASCII characters), and the sector display will be more useful for these.

Program files are written to the screen in lines showing 8 bytes at a time, both as ASCII characters and as hexadecimal bytes grouped as 4 words, but the size of program file fully displayable is limited to about 46 sectors.

The file display in 40 cols may be scrolled by line or page, either by single key or auto-repeat in either direction. Once a record has been read in from disk it is stored to a buffer in low-mem and subsequent access to that record is from the buffer. If a file exceeds the buffer limit of 8K the display halts for user input. If <enter> is pressed the display returns to record #1, but any other key causes it to go into circular mode where early records are progressively erased to make room for later ones. This is indicated by CIRC appearing followed by the new starting record number. Only one file is accepted into the buffer at any time. Once a file has been read into the buffer all or selectable part of the buffer contents may be printed to the FUNNELWEB print device in Append mode after editing of its current default name.

The display is updated in units of a file record which may occupy up to 7 display lines. For program files the unit is a line displaying 8 bytes. Trailing blanks are ignored. The status line keeps track of the numbers of the first and last records visible on screen at any time. The second number will vary rapidly during scrolling. After EoF the number of the last record read from disk is also indicated.

The next set of keys controls the View of the file active in the View screen. The print option is available for whatever part of a file is in the buffer.

<X> and <ctrl X> keys. Pressing <X> causes the display to scroll one record towards the end of the file, getting it from low-mem buffer or from disk as needed. <ctrl-X> sets up a continuous line scroll to the end of the file.

<E> and <ctrl E> keys. Pressing <E> causes the display to scroll one record towards the start of the file, getting records from low-mem buffer. <ctrl-E> starts a continuous scroll back to the start of the document.

<V/A> keys. Scroll the display one page towards the end of the file.

<ctrl V/A> keys. Pressing <ctrl V or A> causes the display to scroll continuously one page at a time towards the end of the file, getting records from VDP buffer or disk as needed. This is a bit faster than line auto-scroll since less screen updating needs to be done.

<Q> and <ctrl Q> keys. Scroll or start auto-scroll of one page back towards the start of the file.

<space-bar>. The space bar temporarily suspends auto-scrolling while held down, or if not auto-scrolling it causes the last manual scroll operation to be repeated.

<key>. To cancel auto-scroll hit just about any key not given a special function. If you have the display stopped just where you want it with the space bar, press another key before releasing the space bar.

<S> key. Moves the display to the start of the file.

<F> key. Moves the display to the finish of the file.

<ctrl 1-5> keys. At any stage each of these sets a marker at the line at the top of the screen. They are all

initialized to the start of the file when a new file is read from disk. Marker 5 is always set to the last record in buffer as it is read in from disk until it is specifically reassigned. In Circular mode markers are set to the start of the file in buffer as they are overtaken.

<1-5> keys. At any stage these keys return the display to the previously defined marker.

When a file has been partially or completely read into a buffer, a new command key becomes available for use, <ctrl-P> for print.

<ctrl P> key. The current file is printed out from the record at marker #1 to the last record before marker #5. This allows parts of a file to be printed out as specified by editing the print device name which may be a disk file or complete pathname .

FUNNELWEB Vn 4.40 DISKREVIEW

Part 2 - Disk snd Sector Utilities

(1) DISK UTILITIES

The Disk Utilities menu is reached from the entry screen by pressing <D> (or <ctrl A> also if no valid directory is present). The menu appears in the box normally devoted to the directory display. Keys <E> and <X> or <space> drive the scroll bar up or down with wraparound and selection is by the <enter> key. The ones provided have been found useful here, and there is room for a few more suggestions.

o FORMAT DISK

FORMAT first requests the number of the drive in which a disk is to be formatted. Cautionary reminders are issued as a checkpoint, as formatting has irreversible destructive effects once started. Then you are prompted for the number of sides (S or D), the density (S,D, or Q for 80-tk drives). <Escape> may be used at any time to back out. Your disk controller and drives of course must be capable of the options chosen. Before formatting is started you are asked if validation is desired after formatting is complete. It is normal and good practice to do this quick but not thorough validation of the newly formatted disk (by reading all sectors). The sector allocation bitmap will be updated to mark the bad sectors, but disks are now cheap enough that it is false economy to use a disk that formats with faulty sectors.

Normally the disk controller DSR subprogram is used to format the disk. Myarc floppy disk controllers with the original 40-Track DSR ROM will format 16 sectors per track this way in double density. DISKREVIEW does NOT allow for this and always writes a disk header showing the normal 18 sectors per track. See the final section on User Defined Flags for details. Myarc FDCs with 80-Tk ROMs appear to default to 18 sector format.

o VALIDATE DISK

This function is invoked either directly from the Disk Utilities menu or else as a follow-on from Format Disk. It attempts to read all sectors on the disk, from #0 up to the maximum indicated in the disk header (Volume Information Block). The information window shows a running decimal count of the sector being read, the last bad sector encountered if any, and the number of bad sectors if any. The <fctn-4> (Break) key is checked before each new sector is accessed, and if pressed it terminates the process,

If there are bad sectors you are asked if these should be marked as used in the VIB bitmap. Normally a disk showing bad sectors should be treated with great suspicion. Save the files on it as desired or possible and reformat it to check whether the errors are permanent. Even if it seems good then, it should be put aside for a while and then re-validated. More sophisticated programs such as DISKHACKER can give more detailed diagnostics.

o RENAME DISK

This one is so simple it hardly needs description. On selection of drive #, the current diskname is shown and also presented as default for the new name.

o SWEEP DISK

This removes all directory references to files on the disk and rewrites the disk headers to look like a freshly initialized disk. The files are not physically erased, and if despite the checkpoint a ghastly error is made, then immediately use the file recovery procedure. The bitmap is just rewritten without reference to the rest of the disk, so the disk should be re-Validated if there is any suspicion of its condition.

When the drive # is entered the disk volume name is read and shown with cautionary messages. If you <Proceed> you will be returned to the Disk Utilities menu after the necessary disk activity.

o RECOVER FILE

File recovery after Delete or Sweep Disk is possible if no subsequent write operations have been made to the disk. Bring up the Directory of the disk before entering this option. The filename to be recovered is requested. After entry the current directory is checked to see if it is already present, and then a search of currently unused sectors is made to find the name. If and when this is found the file cluster map is checked first for possible corruption against the disk bitmap before restoring the file.

It may be that a file descriptor record left over from previous sweepings or deletes is located and processed. To guard against this each recovered file should be checked out before proceeding to the next, and deleted again if found to be corrupted.

o MYARC RAMdisk Utilities

The CALL PART initializing and CALL EMDK drive number setting utilities are supported by means of DSR subprogram access. The only absolute address used is for reading the current EMDK for display, and nothing is written to the RD other than by the CALLs. No range checking is currently done on the number entered for EMDK. No screen editing provisions are currently made to get other than CALL PART(400,80), but enough room has been left in the program that any CALL PART may be installed with the sector editor. Fairware contributors may write for details of how to go about it.

o HORIZON RAMdisk Utilities

The DN and AO/AF subprograms are supported in the form prescribed in the Horizon/OPA Vn 8.14 ROS docs. Disk number change is in the form CALL DN.o.n where "o" is the existing old disk number and "n" is the intended new number. No explicit range or checking is currently done on the entries. CALL AO.x or AF.x turn auto-booting on or off respectively. The ".x" extension is optional if only the first HRD is to be switched.

o QUEST RAMdisk AON/F

The AON and AOF CALLs are provided for the HV99 Quest 512 RAMdisk.

(2) SECTOR and SEARCH UTILITIES

The Sector Utilities are invoked by <I>nspect Sector from the Directory screen. Selection from the menu that appears in the message window is either by number or the first letter of the corresponding entry. Editing by absolute sector or offset into a file is supported. String search is either by file offset or absolute sector, for either ASCII or HEX byte strings. A wild character or wild byte may be set in the search string. Auto string replace is not supported.

<1/S> keys. Lead directly to the sector editor which appears on the sector display screen,

<2/F> keys. Enable a string search in the file currently marked by the cursor in the directory, and string entry follows.

<3/D> keys. String search is now by absolute sector number. See the later String Seardh section in this file.

<E/X> keys. Force the display back to the normal directory commands before scrolling the cursor.

o SECTOR EDIT

On selection of Sector Edit a new screen appears, with a menu of 5 choices appearing in the box at upper right. The current file under the cursor is written up as a reminder. Select by number or initial letter of the option name.

<1/F> keys. Present for editing the File Descriptor Record for the current file under the cursor. Absolute mode is set.

<2/O> keys. Set up editing of the sectors within the current file. The desired offset is entered as a 3 digit hex number starting from zero for the first sector. The maximum offset is shown for guidance. When a sector is displayed in file offset mode both the file offset and absolute sector are indicated.

<3/A> keys. The absolute sector number is then entered in 3 digit hex form before the sector is read.

The chosen sector is presented in ASCII or hexadecimal form in 16 lines of 16 bytes in the upper box. Toggle between these with the <A> and <H> (or <Q>) keys. A short reminder for the editing keys is given in the lower left block, and these follow J. Birdwell's DSKU in the interests of standardization.

The accelerating auto-repeating flashing cursor appears either in the Hex or the ASCII display modes depending on the entry path. When an entry is altered in either display the corresponding change is marked by reverse video. The HEX display accepts only hex digits for editing. A running count of cursor position in hex is displayed, along with the byte under the cursor.

<ctrl E/X> and <fctn E/X> keys. Drive the cursor up or down a line with wrap-around.

<ctrl S/D> and <fctn S/D> keys. Drive the cursor along a line with wrap-around at sector start and end.

<ctrl A> key. Forces the cursor and entry into ASCII mode.

<ctrl H/Q> keys. Force the cursor and entry into HEX mode. The cursor will start at the most significant nybble of the current byte.

<ctrl O> key. When a sector is read in a spare copy is made. This restores the <O>riginal version to undo editing changes.

<ctrl M> key. <M>akes the currently displayed abd edited

sector over as the reserve copy.

<ctrl R> key. <R>ecalls the reserve sector copy to the edit windows. The <ctrl M/R> keys make it possible to transfer sectors from one place to another.

<ctrl N> key. Moves to the <N>ext sector. In absolute mode this is numerically the next. In file offset mode this is the next sector in the file and may skip about over the disk in absolute sector equivalent if the file is fractured (indicated by the asterisk in the directory display).

<ctrl B> key. Moves <B>ack a sector on the disk or in the current file.

<ctrl W> key. <W>rites the sector back to disk at the current sector number on the current disk, after first asking for confirmation. Remember - careless use can destroy your files.

STRING SEARCH

When a string search option is selected a choice is presented for ASCII or Hex string entry. Each entry mode allows specification of a wild card. For ASCII entry this has "?" as the original default, and for HEX search this is byte >00. When the wild-card is entered all occurrences pf the previous wild-card in the search string are changed to the new value. The string is 8 characters or bytes long, and if you do not use the full length, leave the trailing part as the wild-card. Trailing wild-cards are eliminated from the search to speed it up. The function of a wild-card character in a string search is to flag a character or byte of the string to be ignored in making comparisons in the search so that as long as the other bytes match that one in the sector can be anything.

In disk search (absolute) mode the start sector for the search is requested. The search may be terminated at any time by pressing <fctn 4> BREAK. File search shows both file offset and absolute sector numbers as the search proceeds. The HEX search mode allows the search to be forced to even word boundaries. This cuts down on the number of falss alarms when looking for instruction opcodes or word data in E/A programs.

When a match is found in a sector the display switches to the sector edit screen with all occurrences of the matching string highlighted in reverse video.

<1/C> key. Return to directory screen and <C>ontinue search.

<2/E> key. Transfer to <E>dit mode in either ASCII or HEX and absolute or file offset mode as appropriate. The cursor appears at the first occurrence of the search

string. Editing then proceeds as normal.

<3/Q> key. Just <Q>uit the search.

After going to <E>dit during string search <esc> returns to a further selection.

<1/C> key. Continues search at sector after the one last found before going to <E>dit, either by file or absolute.

<2/R> key. Restarts the search from the start, with the existing string as default.

<3/A> key. Abandon the string search function.

User Defined Flag Patch Notes

Several flag or parameter settings are included just after the start of the DR file following the 3 word file header and the initial B @>xxxx instruction. By byte offset these are

>0A -- If you are using a Myarc FDC with 40-Tk DSR ROM, you MUST indicate its presence by setting this word to >FFFF, and >0000 otherwise. >FFFF here will flag Format to use a special direct access 18 sector per track routine for DD formatting.

>0C -- If you have a TI or CorComp or equivalent disk controller set this to >FFFF to bypass some extra disk activity needed by controllers (mostly Myarc) that need to read the VIB of the current disk for correct sector addressing. The universal value is >0000 for distribution purposes.

>0E -- Reserved for future use.

>10 -- GROM entry address for XB loader, >6372 for TI XB Vn 110.

>12 -- GROM address GPL stack entry for interrupt loader, normally >63D3.

>14 -- GROM address for interrupt loader re-entry, normally >6495.

80-Track Notes

Files copied to an 80-track disk with over 1600 sectors (usually 2880) will occupy more space than they would on a DSDD or smaller disk. This is because 2 sectors are allocated at a time. A file will appear as either 1 or 2 sectors longer than on DSDD or smaller disks. One sector of these is due to 2 sectors being allocated for the File Descriptor Record, one of which just goes to waste. If the body of the file has an odd number of sectors then the last one will be allocated 2 sectors. As an example 33 sector EA SAVEd program files chew up 34 sectors on DSQD disks and 34 sector files go to 36 sectors. All 80-track disks are indicated as QD even if single density along the track. The ambiguity is resolved by looking at the disk size.

DISKREVIEW catalogs 80-track disks according to sector allocations, with file lengths shown being those allocated, and not just those occupied by the file. Copies to normal disks will occupy 1 or 2 sectors less space each. All file copies are done using the DSR Direct File Input/Output subprograms to TI specifications. This is not screamingly fast but sector allocation and de-allocation is entirely handled by the disk controller DSRs. No attempts are made by DISKREVIEW to second guess the controller DSRs. The Myarc FDC with 80-track EPROM was used for development and appears reliable at the DSR level. Reports indicate that the Myarc HFDC is NOT RELIABLE at DSR level for file copy operations with 80-track floppies. Use instead the particular version of the Myarc DM tuned for your HFDC.

The Editor SD, or QD show file sizes as their DSDD sizes even on 80-track disks if you need a quick check (sectors used and available numbers are correct however). I am not sure at the moment which is the more useful approach, given that 80-Tk disks are not a common medium of exchange. Please let me know your preference.

Bug Note

A serious bug has surfaced in the use of DR for one hardware configuration here for which no cure is yet apparent. It has 2 Horizon 192K Ramdisks with ROS Vn 8.14. DISKREVIEW uses the officially defined DSR sub-program access for all functions for which they are provided. Failed DSR sub-program access other than sector access as used for directory reading, such as attempted file copy to a write protected disk, causes subsequent sector access to be captured by the other RDs in the system, and sensible results are obtained only for that RD directory. No harm is done to the other drives, as they no longer are even accessed at DSR subprogram level. Once this has occurred the best solution is to reboot the machine. Return to Funnelweb is not sufficient. The condition does not affect file level access. If a fix is found before an update of ROS 8.1x solves the problem, it will be implemented.

DSKU Notes

Many disk libraries use the file annotations created by the DSKUtilities program (by the late John Birdwell). These may now be transferred to the copied files by <N>otes key from the Tagged Files menu. Creation or editing of these will require use of the sector editor. The File Descriptor Record (FDR) of each file on a disk contains the file name, various details of the file, and the map of the sectors occupied by the file in the form of 3 byte groups starting at byte >1A, each describing a contiguous cluster of sectors. Up to 76 such clusters of one or more sectors are possible, but in practice files are rarely found with anywhere near this many fractures. DSKU stores a file comment as ASCII text at the end of this area. This is not a standard TI DSR feature and is not supported by any known DSR. The DSRs navigate the file contents by following the cluster data blocks until a null is found, so if DSR function is to be unimpaired, the clusters must terminate before the comment is reached. If you are not comfortable with sector editing, use DSKU to enter comments and ignore what follows here.

To enter or edit the annotation for a file, go to <I>nspect Sectors and select <1> to display the FDR. The comment is entered in the 35 bytes from >DC to >FE leaving the last byte null. If the 3 bytes from >D9 - >DB are not null, DO NOT alter anything or the file will be corrupted. The desired ASCII text may be entered in these 35 bytes, and the sector then written back to disk. In transferring notes DISKREVIEW checks the previous cluster (>D9->DA), and if it is not null on both source and target files, ignores the file.

jul / 26 / 1991
Funnelweb Farm

FUNNELWEB Vn 4.40 E/A EMULATION

(1) Programmer's Editor

If the Central Menu Screen shows the Assembler side, selection of Editor loads the TI-Writer Editor modified for use as a program code Editor, including a new mode specifically for assembly source entry. All relevant improvements listed in FWDOC/TIWR EDAV still apply. It is generally more convenient with these modifications than the E/A Editor ever was.

(i) The Editor now functions with word-wrap disabled, E/A tab defaults set, and files are saved to disk with no final tab record appended. Incoming tab records are still recognized. The disabling of word-wrap may spare you the distressing sight of 99 sectors of source code reformatting into one giant paragraph.

(ii) If a DIS/FIX 80 file must be written to disk, say in editing of uncompressed object code, use the PF option as F DSKetc instead of SF. This is described in the TI-Writer manual.

(iii) The Source Editor loads C2 as its character set from the E/A system drive. As supplied this contains a larger set than C1, quite suitable for Assembly source which typically has a high proportion of whitespace. If this set is not wanted, copy C1 or character file of your choice over C2. The 5-sector form saves disk space, but the 9-sector files from TI-Writer may be used.

(iv) Pressing <ctrl-B> splits the current line much as <fctn-2> does in word-wrap mode in the TI-Writer editor. The effect is not recoverable except by retyping the blank part of either line.

(v) The <ctrl-O> key toggles a new Editor mode for assembly source code editing (ASMode), indicated by a diamond shaped cursor. In ASMode the alpha-lock may be left up for convenience in typing comments, and as each line is entered into the Text Buffer it is parsed as an assembly source line and converted to upper case where appropriate. Some common typing errors are flagged by a bloop if found. This will also happen on COPY directives but does not indicate any error there however.

(2) Assembler

This loads the E/A Assembler, in a version modified for FUNNELWEB. See FWDOC/SCLL for use of Script-Load with the Assembler as a multiple file MAKE utility.

(i) The filenames remain visible on the screen while the Assembler is executing. Some default entries are provided, but may be deleted or added to as desired.

(ii) If a filename is found in the mailbox by the freshly loaded Assembler, it is written up as the source filename, and with the last two characters altered from /S or ;S to /O or ;O as the object filename, or else just repeated for the user to modify as desired.

(iii) The AID <fctn-7> key calls up the Quick Directory routine at any time during the filename entry process. For this to be available file QF must be present on the disk in the boot drive when the Assembler is loaded. It returns on exit to the start of the entry screen with marked file installed as normal.

(iv) The source filename is passed back to the Editor via the mailbox, and is immediately available for LF when the Editor is invoked after an Assembly.

(v) The object filename is passed back to the Object file loaders, and appears as default.

(vi) Entering any single character for List Device and <enter>ing will cause the full system print device name to appear.

(vii) The assembler files load as a normal Option 2 (E/A 5) program, so other Assemblers in this form may be substituted. .

(3) Program File Loader
~~~ ~~~~~~~~~~~~~~~~~~~

Selection of Option 3 from this central menu screen sets up a selection screen with 8 choices of Load environment shown. The AID <fctn-7> key for QD is available only in the Central Menu and Loaders screens. Filenames entered after the Loaders option is chosen may use disk volume or hard disk patn names and be up to 25 characters long. A full CRU search mechanism as used in SD and DR for sector access is implemented for program file loading, but not for the object loaders which use the normal E/A utilities.

(i) Option 1 emulates the TI-Writer module which hands over control in the GPL workspace, with Text mode set, and with a full set of characters 0-127 loaded from GROM even if they are immediately overwritten by CHARA1. The utility must be in E/A program file format. An attempt is made to load C1 from the boot drive, but no error is issued if it is not found.

(ii) Option 2 sets up a GPL type of environment, as does E/A 5, adequate for most programs normally loaded by the E/A RUN PROGRAM FILE loader from GPL. Handover is in the GPL workspace and the presence of the E/A utilities cannot
~~~

be assumed.

(iii) Option 3 supplies the E/A RUN PROGRAM FILE function for Program files that have been prepared from E/A object files which do use the E/A utilities. Handover is in the USRWSP at >20BA, but this is not written to after file loading starts so that it does not corrupt files which ignore the E/A utilities and load over them anyway.

The utilities are loaded if file EA is present on the disk in the boot drive when the option is selected (not necessary when running from the E/A module). If it is not found a warning honk is given and a discreet little message pops up. GPLLNK has been modified to work with FUNNELWEB (see FWDOC/REPT for details) and the first free address in low memory is unaltered.

(iv) To load assembly program files from cassette, use either the E/A module directly or a special loader program such as CASSLOAD from XB.

The last program file in a sequence may overwrite FUNNELWEB at the top of high memory without trouble while loading, but if a utility overwrites FUNNELWEB either in loading or while running, it should return to the Title Screen on exit or else reload FUNNELWEB.

(4) Object file loaders
~~~ ~~~~~~~~~~~~~~~~~~

LOAD/RUN handles E/A object files, compressed or not. Files which load over FUNNELWEB in the top of high memory, if not auto-starting, will lock up the computer, and may be loaded with Script-Load instead. Otherwise programs written strictly to E/A manual specifications should run correctly. Common sources of difficulty are discussed in FWDOC/REPT. The EA file must be on the disk in the boot drive when this option is chosen or a warning is given (unless running with the E/A module). DEBUG and SAVE from your E/A package both work normally.

(i) Options 5-8 give variations on Load/Run. The normal Load/Run option sets the last free address in high memory (LFHM) pointer in UTLTAB to protect FUNNELWEB as far down as the User List code and/or FSAVE (currently >E98F).

(ii) SCRIPTLOAD (Option 5) is a batch file loader for object files. The Utility pathname with SCRIPT as filename is supplied as default starter entry. See FWDOC/SCLL for details of SL.

(iii) LOW-LOADER (Option 6) allows object files to be loaded starting in low memory at >2000, and then continuing in hi-mem. All normal REFed utilities are available as REFs. The REF/DEF table starts at >E200 and builds down from >E138. Code must not be AORGed above this. FSAVE recognizes Low-Loaded files. See FWDOC/SCLL for details of
~~~

LL.

(iv) If Option 7 is selected the LFHM is reset to the E/A default of >FFD7. This allows the last relocatable autostarting object file to load over FUNNELWEB if necessary, there being no memory contention because the autostart hands over to the program without returning to FUNNELWEB.

(v) Option 8 intercepts the Autostart of object files and the DEF table is displayed as for a normal file load. The LFHM is as for Option 4.

(vi) If a Duplicate DEFinition load error occurs, the offending DEF is displayed after the error message.

The Run part of the Load/Run procedure generally follows E/A conventions except that more information and help is provided along the way. Pressing <enter> with a cleared filename entry (use <fctn-3> or clear to blanks) transfers to the RUN function. This matches E/A module function.

(i) RUN is activated by cursor driven selection from a screen display of the DEF table. The DEF table may be inspected at any time during a multiple file load by pressing <enter> after clearing the filename. The display shows both DEFs, and also now currently unresolved REFs which usually will appear with the first two characters either blank or with unrecognizable patterns. Cursor keys <fn-S> and <fn-D> drive the cursor through the DEF entries on the current page, while PROC'D (or <ctrl-A>) transfers control to the program entry marked by the cursor, and REDO returns to load another file. Pressing <enter> will page through DEF tables larger than a single screen. Pressing BACK, <fn-9> or <ctrl-C>, aborts the load sequence.

(ii) If an Unresolved REFerence is detected when RUN is invoked, the first such REF found is displayed after the error message.

FUNNELWEB 40-COLUMN EDITORS - Vn 5.01

Corrected and updated - Mar / 94

Part I -- GENERAL FEATURES

(1) Introduction

The latest development in the Funnelweb system for the TI-99/4a computer is an extensive rewrite of the 40-column system editor. For better or for worse it remains as compatible with the original TI-Writer and E/A editors as can be managed, with some lessons from experience, since these have been a good match to the capabilities of the basic TI-99 system. The new editor comes in two versions.

o - a minimal version with all basic functions, updating the previous Vn 4.40 issue.

o - a second form which incorporates multilingual features previously available only in an incompatible form in the European version of TI-Writer. It also has All-Characters mode which allows use on screen of the full 8-bit IBM PC character graphics set (as supported by most modern printers), and provides pathname support for hard drive users and right margin adjust in reformattting also. See FWDOC/ED42 for specific details.

These 40-column editors may be used on 80-column systems as well (TI-99 and Geneve) but the standard 80-column version (issued separately - note the new large buffer version using EVRAM 64 Kb buffer does not support 40-col mode) is dual mode and will serve as a superior 40-column editor as well, with much superior SD functions. As in all previous versions it does not depend on any given manufacturer's specific hardware extensions and may be run on a minimal system as long as it can load Funnelweb, the original TI-Writer or E/A modules not being necessary. In particular it is compatible with the Extended Basic module.

This part of the documentation covers general features while the second part reviews command line and control functions in more detail. A third part describes the features and limitations of the alternative version (/AEH) of the editor. Those not already familiar with Funnelweb or TI-Writer should consult TI's original manual (which remains TI copyright material and may still be available from TI in the US at final clearance prices), or else various User Groups have tutorial material or experienced users willing to assist. These documents will concentrate on changes from the original TI-Writer.

The editor is identified internally as Vn 5.00 (5.01 for new /AEH) but loads with no problems from Vn 4.40 of Funnelweb. At this stage in the history of the TI-99/4a there is no longer any grand plan to issue a full update of Funnelweb, but as individual parts are updated they will be so identified.

(2) File Descriptions
--- -----------------

This supplementary package contains editor and auxiliary files, and various utility programs. The character files other than CHAR1 may turn up in archived form. If more than one file fits the category an asterisk is used as wildcard in the name. Entries followed by a "*" have been revised for the Jul/04/93 issue.

```
   ED, EE -- Baseline editor program files.
   ED/AEH, EE/AEH -- Program files for enhanced editor.  See
   FWDOC/ED42 for details.  *
   HELP4* -- Help files in program file format with * an alpha
   character.
   CHAR*1 -- Character files for Euro-Writer mode.
   CHAR* -- Full 8-bit character files for All-Chars version
   (* numeric).
   F4TX*E -- Command text language files for E/W and
   All-Chars.
   INSTALL/ED -- Configuration utility program.
   CONFIG/40 -- Fully annotated configuration file.
   CON/ED -- Abbreviated configuration file.
   4PRINTFILE -- Sample help file prototype.
   HELPMAKE40 -- Help file construction utility.
   CHRCOAL/S -- Typical source file for PC character sets.
   CHARUTIL -- Character file analysis utility.
   FWDOC/ED40 -- This file.  *
   FWDOC/ED41 -- Command and control details.  *
   FWDOC/ED42 -- Enhanced editor special features.  *
```

The ED, EE, FWDOC/TIWR files from your Vn 4.40 Funnelweb distribution disk(s) are now obsolete, as is the part of FWDOC/EASM describing edit functions.

The minimum requirement on your Funnelweb working disk is ED, EE (or another pair renamed to this) for default only use of the editor, with help files as desired. For non-English (E/W or All-Chars) usage, add character and command text files as needed. The language and character files sets are not all complete. This reflects both my limited language abilities and the level of interest in Funnelweb from those parts. German and Swedish are complete, and French largely so. Dutch has some English terms left, but I am assured most Dutch users are familiar with them. Italian is incomplete, and Spanish has not even been considered. It may be that some national language users will prefer to use English command text, in which case the UK file should be copied to the relevant filename. Please contact me if you are able to help with further development of non-English command text files.

(3) Editor Modes and Load
--- ---------------------

The Funnelweb system editor supports two main modes, Word Processor and Programmer's Editor, from the same set of files. The choice is made implicitly from the alternate main Funnelweb selection screens, but may be intercepted and reset at load time. FWDOC/ED41 gives more details of the internal program modes for each of these.

The editor files ED, EE are Funnelweb Option 1 program files, as is permanently configured in Funnelweb itself. This means that the main menu may be used to load any other editor (or other program) in this form and name. Going the other way, the Loaders screen may be used to load the files. If <cr>s etc are to be visible in the word processor, Option 1 should be used to refresh the character set unless it is configured or forced to load another character file.

(4) Character Sets

The baseline editor uses the character set C1 or C2 as loaded by the Opt. 1 loader path, while the enhanced editor gives further possibilities. Whatever the character set, the >1F pattern (the edge char in Basic) is always redefined as the solid mid-line for use as a distinctive screen divider in freeze mode. The cursor of the moment is the >1E pattern. The editor files must be loaded from Funnelweb under a 2 letter filename for character, language, and help files to be found.

The enhanced editor in its special multilingual modes loads new sets of characters, as described in FWDOC/42.

(5) Help Function

A series of help screens may be invoked from the main command line. The help screens are loaded from disk as program files direct to VDP screen memory to speed response, and the number is limited only by the filename series. The filenames expected are HELP4A, HELP4B and so on. A set of useful help screens is provided, with more being added for your selection. Some of these will be useful only with All-Chars mode.

Content of the help screens is completely at the user's discretion, and a sample original file is included. Help files are prepared as a 24 line by 40 column DV/80 file with the Editor and converted to program file format using HELPMAKE40. The help file loader in the Editor starts with HELP4A unless it has been turned off by INSTALL/ED from the CONFIG/40 specification. If one of the series is not found the loading terminates. Pressing <A> pages forward, <Q> pages back, and <esc> exits the Help mode. Re-entry to the sequence is at the screen last viewed on the previous invocation of Help.

The selection of help screens as supplied in this package have the following contents :-

HELP4A -- part 1 of a list of Editor functions with keys to use.

HELP4B -- part 2 of this list.

HELP4C -- summary of Formatter dot commands

HELP4D -- details of Editor extended PrintFile options

HELP4E -- illustration of box drawing characters for All-Chars mode

HELP4F -- other 8-bit characters for All-Chars classified by type

HELP4G -- list of All-Chars 8-bit characters in ASCII order

HELP4H -- E/A quick reference part 1 for TMS-9900 op-codes

HELP4I -- continuation as part 2 of op-code list

HELP4J -- E/A quick reference part 3 for system addresses and color table.

(6) Show Directory Functions

The Show Directory screen is much the same as in previous Editors. The filename and detail display on each line is similar to that in DiskReview except that no check is done on program file type. Fractured files are indicated by an asterisk. Print Directory now uses <ctrl-P> and goes to the DirectoryPrint device. This is initialized at load time to the PrintFile device but maintained separately after that. <P>rotect and <U>nprotect of files now use these more obvious keys. Pressing <T> marks the Display/80 file under the cursor bar in the directory as the LoadTemp file, and pressing <space> marks the Workfile, and pressing <O>riginal restores these to their values at entry to SD. Exit from SD is by <ctrl-=>.

A one way paged file view function is implemented using <V> when a Display/80 file is between the cursor bars, and can scroll through files of indefinite length. The SD screen shows the diskname, sectors used and sectors free counts, and the number of bytes remaining in the text buffer. This includes the effects of buffer encoding and E/W will give a lower figure than 7-bit modes on the same file. The empty buffer value may change with future revisions.

(7) New Load/Save Functions

The Load/Save module now has code which performs extensive validation tests on incoming tab records from any mode into any mode. Loading and saving of text records bypass DSR search and go directly to the opened DSR for improved speed. A line count is at the upper right of the screen. Changes under user selection are in the option codes for PrintFile.

M -- sets PF to output the file in DisFix/128 TI file format with MS-DOS end of line <cr><lf> separators and <ctrl-Z> end of text marker.

U -- does similarly for Unix format with <lf> separators and <ctrl-D> at end of text. M and U both cancel the L option for line numbers.

P -- if a printer start-up control code sequence has been installed this will be sent to the print device before any text records.

Q -- if a printer reset control code sequence has been installed this will send it to the print device after all text records have been output.

A -- opens the DV/80 output file in Append mode. This will help editing or construction of very long files.

There is no provision for external files in the M/U formats to be read in directly, and external conversion programs will be needed to produce DV/80 files first for loading by the Editor. For information on the other PF options carried over from earlier editions, consult the help file for a brief account, or else the TI-Writer manual.

(8) Configuration of ED

The program INSTALL/ED (program file - FW option 2) allows a range of initial options to be installed in ED from a DV/80 text file. CONFIG/40 is such a file, and is its own documentation. Keep for reference, but a cut down version such as CON/ED will do just as well. The baseline editor ignores all references to language selection.

(9) Printing

The usual Funnelweb modification of the Formatter, or other alternatives such as the RAG Formatter, will work as before with 7-bit files. Printer codes to set national character sets will depend on your particular printer. All-Chars files should print directly from the Editor to printers (most nowadays) that support the PC character graphics set. A rewrite of the Formatter would seem to be in order at this stage, using the existing TI-Writer manual as base specification in the absence of original source code, but life seems just too short with too many other things to do.

The existing Funnelweb Formatter functions with the following enhancements over the TI original.

(i) The printer device-name is preconfigured in the main FW/LOAD program by CF/CG and may be changed to suit your convenience.

(ii) The Formatter will display the filename last used or marked. If it cannot locate a name then the utility pathname set in FUNNELWEB with CF/CG becomes the default.

(iii) The <fctn-9> key now returns directly to FUNNELWEB's central menu screen. The Formatter does not need to reload from disk if reselected immediately.

(iv) Pressing AID <fctn-7> invokes the a directory routine similar to QD. File QF is loaded if present on the boot disk when the Formatter first loads for this to be available. File marking is active, but is not apparent until the workfile name is edited - <enter> is not enough but even a cursor movement will do.

(v) The Formatter may now have 4 disk files open at the same time.

(vi) The FO/FP files load purely as an Option 2 (E/A 5) program, and another Formatter program in this form may be substituted.

(10) Fairware Notice

This program is distributed as FAIRWARE. Consequences of its use are entirely the responsibility of the user. The files as sent out are fully functional, unlike the frequent and unpleasant custom of "cripple-ware" in the PC world. The usual FAIRWARE conditions remain in force for this extension of the FUNNELWEB system. If you don't think it is worth supporting or don't bother to, then be honest and don't use it, but at the very minimum please show your appreciation of the free trial by passing it on to others who may.

(11) Final Words

Please report any bugs found, or send suggestions. The existing functionality is very close to the limit that can be squeezed into standard TI-99 hardware. If something is added other things may have to give.

WARNING - the Myarc HFDC when used as a floppy disk controller is known to corrupt files when multiple fractures occur. At least some DSR ROMs for 80-track operation of Myarc FDCs are known to get wrong last sector allocation in files. The Funnelweb Editor cannot correct for either of these equipment faults.

FUNNELWEB 40-COLUMN EDITORS - Vn 5.00

PART II -- COMMAND and CONTROL

(1) Editor Modes

Each of the Word Processor (W/P) and Program Editor (P/E) has two sub-modes, toggled by <ctrl-O>. In W/P these are the familiar word-wrap mode (solid cursor) and fixed mode (hollow rectangular cursor), and are essentially the same as set out in the TI-Writer manual to which you are referred.

In P/E mode the initial state is a modified and locked fixed mode with hollow cursor, and is set up for writing source code in languages such as c-99. Tabs are initialized to E/A editor settings. Word-wrap is disabled to prevent accidental reformatting of source files into one giant paragraph, and <cr>s are never written except by special character mode. <Ctrl-O> toggles to a new ASMode (with diamond cursor) for writing assembly source code. In this mode each line is partially parsed as assembler code before it is stored in the text buffer when the cursor leaves the line. The label, opcode, and operand fields are automatically up-cased as required by the assembler and some checking is done for common errors such as unmatched quotes or "." instead of "," in the operand field, and non-alpha characters in opcodes.

(2) New and Updated Editor Command Line

Several new command line 2-letter entries have been introduced in various categories. An important and the most obvious change to command mode is that text may be scrolled by line or page using the normal set of up/down scroll control keys. This allows the text to be inspected anywhere during command line entries, so that line numbers do not have to be remembered for large Copy/Move/Delete operations. The new entries are specified here by their English language version.

<T > -- for Tabs is not strictly new, but now brings up a second command line which asks TABSETS (1-3)? and indicates the current setting as the default entry. Tab records are saved with files by the Word Processor and not by the Program Editor, but are recognized by both.

<H > -- for Help mode brings up a series of help screens which are loaded from the Funnelweb disk with paging between them by <Q,A> and exit by <ctrl-C>. See FWDOC/ED40 for details of preparing Help screens.

<QQ> -- for Quick Quit back to F'Web. The editor maintains

a "file-edited" flag, and if any text entry has been done since loading or saving the current file, a reminder to save the current work first will be issued. This warning also operates before Purge.

<LT> -- for LoadTemporary file. The temporary loadfile name may be entered directly, or marked in SD with <T>. This allows for inserting all or part of external files into the edit workfile without disturbing its name.

<DP> -- for set showDirectoryPrinter name. This allows the device name used by <c-P>rint Directory in SD to be preset to something other than the PF name. It is initialized to the PF print device at load time.

<MK> -- for MarK position in file. This sets a marker after line number entry, or else enter this with <ctrl-M> at the current top line, which may be scrolled to any line in the workfile while still in CMD mode. <fctn-;> in Edit mode is an alternative method.

<WC> -- for choice of WildCard character for use in FS/RS search strings. This initialized as the "*" character.

< > -- a blank CMD line. On the main CMD line this returns to the Edit mode at the original exit point.

<number> -- from the main command line a number acts like a Show lines command. "E" for EoF is not recognized in this direct return as a letter may conflict with other commands. You can just use a big number, say 2222, instead of <E> but it is easier to use <S > for Show line.

Some control key presses now have new special functions in CMD mode, and mostly were of no function before. Where the new function also applies in Edit mode it will be listed in that section. For a summary of all key functions see the help screens supplied as HELP4A and HELP4B.

<ctrl-M> now writes the current top of page line number at the cursor position on the command line in insert mode. If you must have <cr> on the command line use <ctrl-8> or special character mode.

<ctrl-1> exits from command mode to the current top of page. It has the same effect as <c-M> followed by <enter>.

<ctrl-2> exits from command mode to the departure point from edit mode.

(3) Find and Replace String

Find/Replace String commands now take up to 3 numbers ahead of the string entry. Two numbers give the start and finish column for the search. For 3 numbers or 1 number the first or only number is the number of match occurrences to skip before stopping. This is similar to the E/A editor. In case you had not noticed, RS always worked like this. Also when no more matches are found, BOTH FS and RS give an audible bloop and stop where they are. The start position for the search is resumed with <ctrl-O>.

Any non-numeric character may be used as delimiter, so that /ABC/defg/ or -ABC-defg- or aABCadefga as RS string entry will all search for string ABC to be replaced with string defg. A wildcard character, set by <WC>, can be included in the search string. The search procedure ignores the character in the text line corresponding to each wildcard occurrence in the search string. Neither delimiter or wildcard can be a regular part of the search string.

(4) New Edit Mode Functions

Changes have been made to the edit control keys so that many functions are available from the left hand in a compact block without stretching. Some keys were already in place such as the cursor diamond <ctrl-ESDX> which duplicates <fctn-ESDX>, and <ctrl-C> as shadow of <fctn-9>.

<ctrl-Q> pages towards the start of file (<fctn-6>).

<ctrl-A> pages towards the end of file (fctn-4>).

<ctrl-Z> places the cursor after the end of the current line and is no longer the alternate Oops key which remains on <ctrl-1>.

<ctrl-H> shows the first page of the file.

<ctrl-J> shows the last page of the workfile.

<ctrl-B> breaks the current line at the cursor in all modes, but does not enable <cr>s with <enter> in word-wrap mode. In W/P fixed mode it replaces <fctn-2> which splits the line only in wordwrap mode.

<ctrl-R> rejoins lines that <ctrl-B> has broken. More precisely in wordwrap mode in the W/P it remains as an alternative key to <ctrl-2> as reformat. In all other modes it inserts the contents of the next non-blank line (blank includes paragraph break lines with <cr> only) into the current line at the cursor position. Leading spaces and trailing spaces and <cr>s are trimmed from the inserted material. If the effect displeases, just use Oops <ctrl-1> immediately. So there is now a way in the various fixed modes to insert material into a line without having to retype it.

The redefined <ctrl-H,J> no longer duplicate <ctrl-6,4>. In Program Editor <ctrl-4,6> search, instead of for <cr>s marking paragraphs, for either asterisks "*" in the first column as marking assembly comment lines, or for the c-99 comment delimiter "/*" at the start of a line. Strictly speaking the search is for the first non-blank line following the target item. This substitutes jumping between comment lines in source code for paragraph jumping in W/P text. It also removes the annoyance of time consuming traversals to the start or end of source files if these keys are accidentally pressed in P/E mode.

Some minor changes have been made to improve safety and convenience in editing. <Ctrl-N> in Edit mode now inserts a New line to match usage on PCs, as in Borland editors. Some other new functions are provided.

<ctrl-F> freezes the bottom part of the screen under a solid line drawn across the screen on the line below the cursor. Horizontal windowing does not shift the frozen part in 40-column mode.

<fctn-;> sets a bookmark for the line at the current cursor line. It is equivalent to MarKing in command mode.

<fctn-=> effectively does a Show Line with the currently marked (<fctn-;> or MarK) line at top of screen. It is reasonably intelligent in the face of changing workfile contents, and if confused reverts to line #1. It has been disabled as the system reset key combo.

<ctrl-O> returns to the Original line after some operations such as <fctn-=>, RS, and FS.

<ctrl-M> in the Program Editor only, inserts a blank line following the current line and places the cursor on the new line under the first character of the current line. If this line was blank the cursor stays in its current column. It retains its New Para function in word processor mode.

<ctrl-2> in the Program Editor only, deletes the current line if and only if it is blank between the current left and right margins. This makes it a lot safer for deleting a bunch of blank lines than <ctrl-3> which can do real damage in careless moments. It remains as Reformat in W/P word-wrap mode (solid cursor).

(5) Performance Enhancements

This update retains all enhancements of recent Funnelweb editor versions over the TI original, and those not mentioned elsewhere in these notes are listed here.

(i) Text buffer capacity in 7-bit modes is increased by improved encoding, the degree depending on buffer contents.

(ii) The color selections using <ctrl-3> are the 10 configured in FUNNELWEB using CF/CG.

(iii) The printer device-name is read in from the main program and used as default for PF and directory <ctrl-P> printout in SD.

(iv) The current Funnelweb system workfile name is used as LF and SF default. At the initial Funnelweb load a default workfile name may be configured with CF/CG into FUNNELWEB. If left blank the default utility pathname or the pre-existing filename will be set. If your system has 32Kb RAM in battery backed form, as on some RAMdisks, it may well survive power cycling if not otherwise wiped out by programs such as MENU on HRDs. Use FW as your auto-boot program on HRDs.

(v) The <fctn => system Quit key (<fn-ctl-=> in AVPC machines) remains disabled at all times while in the Editor, including SD.

(vi) The text buffer manager routines have been completely rewritten and screen painting has been speeded up slightly to give "crisper" screen scrolling. Delay in word-wrap has been reduced so that there is less problem with loss of keystrokes, and Reformat is faster.

(vii) Delete Lines is greatly speeded up, particularly on freshly loaded files.

(viii) Copy Lines is now very rapid, and does not leave partial copies if 'Text Buffer Full' would occur.

(ix) Move Lines is now instantaneous and no longer can cause the 'Text Buffer Full' condition. It now merely shuffles line numbers instead of copying and then deleting. I have a suspicion TI's programmers were following big company rules for structured programming, with the usual result of bloated and slow code.

(x) A right margin warning beep has been incorporated as a beep occurring 5 spaces in from the right margin during typing.

(xi) Alpha case conversion is provided, in either Edit or Command modes. <Ctrl-;> converts a lower case letter under the cursor to upper case, and <ctrl-.> below it on the keyboard converts upper to lower case, with auto-repeat.

(xii) The End-of-File message has been replaced with a full width ruler line which shifts with window and line number selection.

(xiv) The margin release key <ctrl-Y> now gives full release on both left and right margins.

(6) No Longer Supported

The Recover Edit <RE> function from the command line is no longer included. This had lost most of its original function in Funnelweb as the text buffer contents are overwritten on return to Funnelweb, unlike the original TI-Writer which returned to GPL code in the module GROM. As noted above under <QQ> Quick Quit, a warning is issued if the file contents have been added to since last loading or saving. This matches common practice on PCs. The Oops line recovery function remains unaltered on <ctrl-1>.

APPENDIX - Updates and Bugfixes

Updates are mentioned only briefly. For details of new or improved features, see the relevant sections of these document files.

o May 1993 -- Initial release

o July 1993 -- Second release

Changes :

(1) Program files provided for basic editor, and for an enhanced version, which now gives the Euro-writer support.

(2) Keys in SD in both editors altered so that <Q,A> replace <B,N> for page up/down. <c-A> no longer exits SD in normal ED. System consistent delay loops used in SD also.

(3) Command language text auxiliary files removed to special request status.

Bug-Fixes :

(1) Reloading of FW/LOAD from enhanced ED has been reworked and corrected.

(2) <D>el files in enhanced SD now returns correctly, and <c-P>Directory print-out corrected in both versions.

o March 1994 - third release

Changes :

(1) Right Margin Adjust added to /AEH version, using <ctrl-R> as alternatve reformat key.

Bug-Fixes

(1) RS fixed in /AEH to match correction in fixed baseline model.

FUNNELWEB 40-COLUMN EDITORS - Vn 5.01

Corrected enhanced - Mar,Apr / 1994

PART III -- ENHANCED EDITOR

(1) The Alternative Editors

This package of files for the Vn 5.01 Funnelweb 40-column Editor contains a further set of editor program files under the modified names ED/AEH, EE/AEH. The one you choose to use should be renamed to ED, EE for use on your Funnelweb work-disk if you require language capability, or the pathname facility. Alternatively, either or both sets may be given another 2-letter name set to load from another central menu slot, or from a User List. These files provide all functions of the baseline 40-column editor except that right margin adjust, TI EuroWriter, PC character graphics capability, and pathname functions in SD, have been added. These have until now been available only in the 80-column Editor. A minor disadvantage of the enhanced editor is that the program files take more room on disk.

The price paid for this increased functionality is that the Funnelweb kernel can no longer be stored in VDP during Editor operation for instant reappearance on exit, but must be reloaded from the Funnelweb boot-disk. This is handled automatically, but will cause noticeable delay from floppy disk. Users of Horizon or similar RAMdisks will hardly notice the change, and it should be quite rapid from hard disk. You will however lose any on-the-fly customizations, such as marking of program or object file defaults for the Loaders screens. If neither FW or LOAD are found, it will return to the title screen. Operation is similar to that used on the earlier FW version of DM-1000 except that the original boot path is assumed always, with no option to change it. We recommend that users with 80-column capability use the original small buffer 80-col Editor instead, as it supports all features of both 40-col editors and more, and can display in 40-col mode also (but not the new 64 Kb EVRAM buffer version). This is still on the 80-col or combo disks as files E8, E9.

(2) Language Capability

TI released in Europe in 1983 (in Germany at least) a multilingual Version 2.0 of TI-Writer which supported the range of languages implicit in the TI-Writer module selection screen. We will refer to editors of this style as Euro-Writer or E/W. Unfortunately E/W writes Tab records to file which are fatally incompatible with the original USA issue of TI-Writer. It also had a whole range of auxiliary text and character files, and a new Formatter with special transliteration files for the new characters.

The new Funnelweb Editor supports both the original TI/Wr and E/W with selection at load time, either preconfigured (see later) or from the selection screen. The file loader handles all existing tab records (TI-Wr, E/W, F'Web) transparently. The user selection screens are brought up by pressing <space> as the program starts. First choice is between Word Processor and Program Editor. The next choice is from 3 options.

<1> Default 7-bit, in which no further character or command files are loaded.

<2> National 7-bit, which is standard TI-Wr, but loads national command and character files. This will be useful in languages and applications which can coexist with a modified 7-bit character set, and is available in both enhanced editors.

<3> All-characters, which supports the PC character graphics set as implemented in most modern printers. It is described in detail in a later section.

<4> TI Euro-Writer, which apart from some redefined normal characters, allows entry of various modified versions of vowels, using keys <fctn-,./> and <ctrl-/>. These are encoded as ASCII 128 to 167 (>80 - >A7). You will need your Euro-Formatter and transliteration files to handle these E/W files correctly in printing if they contain modified vowel characters. These are NOT provided with this package.

The next option box allows selection (1-8) of the various national languages. Option 1, Default, is the base line option with no further file loads. Modes (2-8) use various auxiliary files. National 7-bit and E/W mode load text/command files F4TX<A to G>E, and the TI E/W files CHAR<A to G>1. Loading an E/W file into the 7-bit Editor modes may corrupt the file as the MSbit is stripped from all characters. If in doubt, load into E/W mode.

(3) Euro-Writer Operation

Some new key functions are used in Euro-Writer mode, on keys which were not used in the original TI-Writer. They are enabled when E/W mode is configured or selected at load time.

<fctn-,> in Euro-Writer mode only, modifies the normal vowel under the cursor to one with a circumflex accent. Vowels so modified must be retyped to normal form for changing the accent. Some of the modified forms may already be available in some national character files as alternative versions of regular 7-bit ASCII codes.

<fctn-.>,<fctn-/>,<ctrl-/> similarly apply umlaut, grave,

and acute accents respectively.

All other functions are as for the baseline editor, except that the tab records written when in specific E/W mode are as for the European TI-Writer and so incompatible with most previous editors.

Printing from this version will require the TI Vn 2.0 European formatter as the special characters are not widely supported. No work has yet been done on interfacing the the E/W Formatter, and we do NOT intend to supply it or its auxiliary files. For the moment change the word in your copy at >30 in the first sector of FORMA1 from >130A to >100A so that you at least can use it with Funnelweb or E/A. Edit the drive # and language letter in the string DSK1.TXTFA in sector >0D, and at >20 of sector >0E change >D800 to >9800 to disable the language selection path from the TI/Wr module.

(4) All-Characters Operation

All-Chars allows use of the full 8-bit IBM PC character graphics set as supported by most modern printers, and accessed by printing directly from the Editor via PF (a Formatter version to support this is not available at this time, and may well never be done). New character files CHARx are needed on the Funnelweb work-disk. The command text files remain the same as for the 40-column E/W mode, at the cost of a little redundancy in each. National 7-bit mode remains as for the E/W model.

This mode may be configured in, or selected from the load time selection screens. A full 8-bit character file, CHARx where "x" reflects the language choice is loaded, along with the command text file in languages other than English. The 7-bit characters in the language files are as for E/W and the 8-bit set (ASCII 128-254) are as in the CHAR1 file. The ASCII control characters are represented in TI-Writer form rather than the IBM code page 437 format. At this time files are as for TI Euro-writer in the 7-bit component. In languages which use 8-bit EuroWriter characters in command text files, these text files may need to be modified (either from source or using a sector editor).

In All-Chars mode the text buffer encoding used is as for the EuroWriter mode in that Editor, and buffer capacity will be smaller than for 7-bit modes. Tab records may be configured to be of either original TI-Writer or Euro-Writer form. Remember if you are sending files to other people that only the Vn 5.0x Funnelweb editors will handle both forms gracefully. Loading an All-Chars file into normal mode will in general corrupt the file by stripping out the extra bit set for IBM graphics characters, and in places where this is not done as in Help screens intended for All-Chars advice, random patterns will appear instead.

The 8-bit character mode is toggled by pressing <ctrl-,>, but in the 40-column version there is no specific indication that this mode is in effect except by the characters typed on the screen. The space bar in this mode will result in a character on the screen, and the right shift keys should be used for spacing right. Entering command mode always cancels the 8-bit mode, but it may be reset there. All characters may be entered (ASCII 0-254), but ASCII 255 (which would be <fctn-V> in this mode) is always replaced by a regular space (ASCII 32) when a line is transferred to the text buffer (this is because >FF is used as a flag byte in text buffer encoding). This key may be used to correct accidental <space> entries in this mode, but the change to a blank from the marker arrow character will not be immediately apparent on screen.

Some tools are included to help construction of char-files. CHRCOAL/S is source code for constructing these from character pattern data. It also serves as an example of how a program written to run under Funnelweb can call system services such as QD directory. CHARUTIL extracts source data from existing char-files. These utilities are included because I found existing public domain char-set tools quite inadequate for practical use.

(5) Enhanced Show Directory

The SD function has been extensively revised for the All-Chars Editor. Individual directory pages now contain up to 16 files, and the current work-file and temporary load-file names are shown explicitly. The drive selection is now entered after the SD screen appears. A second one-time page scrolling View mode is available from <W>, which shows only the first half of each line on the 40 column screen. Particularly with program source files, this may well show enough information for View purposes with less on screen confusion.

(6) Enhanced Pathname Support

The enhanced editor offers the same pathname support functions in SD as in the 80-column Editor. <HD> from the command line brings up a pathname for editing. This is initialized at load time as either the Funnelweb utility path or as installed from CONFIG/ED. The pathname entered must end with a "." or it will be ignored. SD then reads the directory associated with this path as described for <0> below. The command line text in the separate command text files does not contain any explicit mention of "HardDisk", to keep these common between both 40-col versions.

In HD pathname mode all disk access is at the DSR file level, so fracturing of files cannot be detected. The protection status of files is indicated but cannot be altered, as the sub-programs other than for floppy disk DSRs are not currently supported. Marking and viewing are file level operations and are available as usual. Special behavior for the pathname mode is associated with several keys.

<0> -- as the disk number reads the Internal, Relative 38 catalog pseudo-file for the pathname as configured or as

last entered by <HD> from the command line. This entry uses an assembly language version of the standard Basic program for reading the catalog file as specified by TI for any drive or as extended by Myarc to directories on hard drives. This mode operates at file DSR level and a directory so obtained does not indicate fractured files. File protection is shown, but nothing can be done about it, as alteration may require sub-programs not defined for standard floppy disk DSRs. Marking, deleting, and viewing function normally. The sectors used/free display may well be nonsensical for hard disk directories, but is retained for occasions when it is correct.

<ctrl-O> -- in pathname mode reads the parent directory of the currently displayed sub-directory. The sub-directory name is also trimmed off the pathname stored in the <HD> name buffer.

<space> -- still marks the Display/80 file under the cursor bar as the current workfile, as used for LF and SF. In addition in pathname mode, if the cursor bar is on a Sub-Dir entry, it will cause that sub-directory name to be appended to the existing pathname, and the catalog is generated for that sub-directory. The augmented pathname becomes the current pathname as stored in the <HD> name buffer.

(7) Right Margin Adjust

--- -------------------

This provides an alternative path for <ctrl-2> reformat in word-wrap mode, in which each line is adjusted as its content is decided by the reformat so that an even right margin is obtained, for tidy appearance when printed directly from the Editor, without requiring the Formatter or painstaking adjustment of each and every line. The key <ctrl-R> which formerly shadowed <ctrl-2> now adds right margin adjust to each line after the normal reformat has decided what is to be on the line. This paragraph as you see it on screen is an example of its function - it makes more sense in full width texts. The functionality is exactly the same as in the new large buffer 80-col editor.

Apr/20/94 Modifications

o Changes -- Added features are right margin adjust, line hit indicator for RS All.

o Bug fixes -- Purge after Quit lockup, replace string corruption and lockups.

Tony McGovern
215 Grinsell St.
Kotara, NSW 2289

Australia
(049) 52 3162
e-mail - phpamcc.newcastle.edu.au
Delphi - GLOBAL01 on TI Net

Apr / 20 / 94

FUNNELWEB Vn 4.40 LOAD INFORMATION

(1) First Things First

Before doing anything else, make a working copy from the distribution package. You may need to unarchive varying amounts of it, depending on the particular form of the package you have. Keep the original safe as a master copy for backup and for passing on. All files may be un-archived and/or copied in the normal way. Copy the frequently used short system files on to the working floppy disk first to speed access in use. The usual format for distribution is as DSSD, in archived form, or DSDD unarchived. If you are re-archiving it is recommended that each group of files be copied to a clean disk and archived as "all files" on the disk.

If you have only SSSD disk drives, and do not wish to use the E/A functions, then QD, QF, C1, UL, LOAD, ED/EE, FO/FP, DR/DS should be included, with loader utilities EA, SL, LL etc as required by your Menu and User Lists. SSSD users will find it convenient to have different partial system disks for various purposes.

(2) Loading FUNNELWEB

The FW file loads from E/A 5 and may be renamed UTIL1 for easy loading as DSK1.UTIL1 from E/A if RUN PROGRAM FILE is selected and <Enter> pressed with no file-name. Under name FW it gives a easy CALL and/or auto-boot file with the Miami UG ROS for Horizon style RAMdisks. The Utility option of TI-Writer also provides auto-load of DSK1.UTIL1. The file can be loaded under any name if desired.

When FW executes it loads a character set equivalent to that in C2. It can be configured by CF/CG to go directly to either DISKREVIEW or User List UL unless it finds the <space bar> held down in which case it goes directly to FW. If you are loading from a Horizon RAMdisk with Vn 8.1x ROS, having DR as a CALL name will give a particularly rapid load (and if DR so set up, then this one will be auto-loaded rather than the DR on the boot disk). DR also has its own display of the Central Menu.

When FW loads directly to the Central Menu it starts with a FAIRWARE advice screen, which will time out if no key is pressed. The space bar will hold the screen in place until released. <escape> as <fn-9> or <ctrl-C> will bring up and hold a second screen.

With XB and Myarc XB-II just auto-load DSK1.LOAD. LOAD is a hybrid of Extended Basic and machine code and must only be edited by means of the CF/CG utility program, which is largely self-prompting in use, and has many built-in help screens.

Direct editing must NOT be attempted. Use a disk manager to transfer LOAD from disk to disk (it can be reSAVEd unedited with TI XB but NOT with Myarc XBII).

Object file LDFW auto-runs from any compressed object file loader. CT8K/O is a utility which installs FUNNELWEB in devices that provide 8K RAM in the cartridge space.

(3) Hard Disk Operation

LOAD/FW may be configured to operate with a pathname of form WDS1.FWB. from a subdirectory on the WDS1 hard disk containing all the centrel menu (2-letter name) utility files. This will free the HFDC file DSK1 emulation for use with various programs such as disk directory routines that use sector access. The utility disk default name is now treated as a pathname also. Other pathnames for RAMdisk access are also configurable.

If a Horizon style autoloader is not available, place copies of the configured LOAD and FW (as UTIL1) in the DSK1. directory on the hard disk or else on a Horizon to allow easy loading from XB, EA or TW modules.

(4) Boot Disk Tracking

FUNNELWEB either from XB or as FW (under any name in form DSKn.yyy), with hard pathname configured OFF, locates the drive number "n" used to load it, and writes this as the system drive number. This means that the FUNNELWEB disk may be used from any drive if booted from that drive. From loaders which first ask the user for drive numbers, the ones supplied are used.

(5) Source Disk Specification

LOAD/FW has an escape hatch which disables boot tracking and allows the system drive or pathname to be specified directly. This is mostly intended for Horizon style RAMdisks when direct CALL access to FW or LOAD is used, particularly in auto-booting, or when separate source disks are specified for TI-Writer and E/A system utility files, as may be done in systems where a single SSSD or DSSD capacity is insufficient. CF/CG allows setting of these drive numbers as a single character "n" in the form DSKn. for each source disk. Forms such as RD. or WDS1. may be handled via the hard disk pathname option and this overrides the other choices.

The secondary system drive number is for the E/A side of the Central Menu. If separate disks are specified, FUNNELWEB looks for the system files in the appropriate drive first before checking the other one. Character files C1 and C2 (formerly CHARA1/2) are the only system files which must reside in the particular drive. With this exception the system files can reside on either disk but fastest loading occurs when they are correctly partitioned. The drive numbers may be edited from the Central Menu screen after <ctrl-C> as also the character filenames.

(6) XB Selection screens

After auto-loading from XB a title screen appears while QD is loaded. The title screen will time out if no key is pressed first, to a screen with 18 choices, 9 by number and 9 by letter. The first 3 are preset internal paths and the remaining 15 are configurable. The first two choices, TI-Writer and Edit/Assm each lead to the Central Menu screen with the corresponding mode set. When these paths are taken the XB environment is destroyed and cannot be restored except by rebooting XB, which takes just 3 easy keystrokes, or from DiskReview which can restart TI XB.

(7) XB User's List Selection

The remaining 15 entries have names which are entered in the LOAD program by CF/CG. On selection of one of these 15 user list items FUNNELWEB loads the program configured into that option. This may be an assembly program, XB RUN of another XB program, or just return to the XB command mode. LOAD may be reRUN from the command line to allow recovery fron XB RUN errors without complete reload from disk. This path should work with any dialect of XB. Some entries have been predefined by way of convenience and example.

At this stage the set of XB INIT utilities is still available for either XB programs or for assembly program files (options 1 or 2). These have been augmented by a DSRLNK (BLWP vector at >24F4 with function as defined in the TI-Forth source code), and a GPLLNK (BLWP vector at >24F8).

(8) Universal key functions

(i) Screen colors

On almost any screen which invites selection by number, pressing 0 (zero) will cause the screen colors to cycle through through the 10 configured color choices.

(ii) Quick Directory

The AID key, <fctn-7>, will call up from file QD a paged (by single B/N or <ctrl-E/X> keys) disk directory routine with some of the functions of that used for SD in the Editor. A very similar function exists within the Formatter and Assembler using file QF instead. File QD must be on the FUNNELWEB boot disk, and is an Option 2 program file. The QD file will be reloaded if necessary from AID. Pressing the <space-bar> will cause the filename indicated by the cursor to be written to the mailbox as workfile name if a Display file, as the program default if a program file, and to the object default if a D/F file. Any file may be deleted. QD indicates file lengths on 80-Tk disks as though they were on normal 40-Tk disks.

(iii) QUIT <fctn-0>

The system QUIT key is enabled at all times when keyboard entry is called for by FUNNELWEB itself. Each utility determines its own response to QUIT.

(iv) BACK/<fctn-9>/<ctrl-C>

In FUNNELWEB itself <ctrl-C> may be used instead of <fctn-9> (BACK) as the <escape> key just as in the Editor, and <ctrl-A> usually substitutes for <fctn-6> (PROC'D). Under some error conditions these alternatives may not be available. From the Central Menu screen <esc> brings up a Y/N choice to Quit. This is the recommended exit from FUNNELWEB if the machine is not to be switched off immediately as it resolves any pending workfile name transfer to the mailbox. If response is not "Y" and <enter>, a Reset screen is entered. The character set filenames may be edited, the system print device changed, and the system drive numbers revised if the hard pathname is not enabled.

(9) Central Menu Screen

The prime focus of the program are the Central Menu screens which offer 8 choices on the TI-Writer entry (hit the space bar if necessary).

```
1 TEXT EDIT
2 FORMATTER
3 DISK UTILS
4 DM-1000
5 ARCHIVER
6 ,,
7 DSKU
8 USER LIST
```

The normal TI-Writer Editor is loaded from this screen. Entries 4-7 are completely configurable, and may be any assembly program (with a two letter filename) that FUNNELWEB will handle, including Script-load files. Automatic search on both primary and secondary disks occurs only for program files.

USER LIST files are of Opt 2 type, and allow a tree of choices as big as your disk system will allow. Each presents a third screen of user entered options. See FWDOC/SCLL for a fuller discussion of User Lists and Multi Lists, and also the CF/CG instructions in FWDOC/UTIL. Choice #3 is set up with a List file D1 as example, grouping a collection disk-related utilities. UTIL1/FW may be configured so that the User List boots up first. This allows application disks to be made up without the FUNNELWEB machinery being immediately apparent.

A second menu screen comes up from other loaders or may be toggled with the first screen by hitting the space bar. This screen contains the programmer's workbench.

```
1 PROGRAM ED
2 ASSEMBLER
3 LOADERS
4 c-COMPILER
5 DISKHACKER
6 LINEHUNTER
7 ..
8 DISKREVIEW
```

As with the other screen, Options 4-7 are completely configurable. As issued Option 4 loads the c99 REL4 compiler if the appropriate files are on your working disk. Option 6 LINEHUNTER is a search utility to help assembly programmers use the FUNNELWEB system (see FWDOC/UTIL). Option 3 enables the program loading screen (see FWDOC/EASM). The Editor is TI-Writer modified for source code editing. If dual system drives are specified it is not necessary to repeat ED/EE on the secondary disk if the extra loader delay can be tolerated. See FWDOC/EASM for E/A details.

The Option 8 entry DISKREVIEW loads or re-enters a new powerful directory, filereader, sector editor, and loader utility DR which has both 40 and 80 column forms. This is described in FWDOC/DR40 41 or FWDOC/DR80 81 82.

~~~~~~

FUNNELWEB - PROGRAM SERVICES for ASSEMBLY USAGE

~~~~~~

Version 4.40

The FUNNELWEB system is held together by a reference block at the top of high memory. This has grown somewhat haphazardly over the years, but at this stage of the history of the TI 99/4a computer, plans for complete tidying up and revision have been shelved although almost all functional improvements, and more, previously intended for that grand revision have now been incorporated in Vn 4.40 of FUNNELWEB.

The information and facilities available include BLWP vectors for a variety of services, indirect BL routine pointers, and various data, pointer, and flag words and bytes. Specifications of various useful BLWP and BL routines and associated data/flag items follow. Some others, mostly data and flags, are specified in FWDOC/REPT. The XOP instruction is poorly supported on the 99/4a, and is left for application programs to use.

(A) BLWP UTILITY ROUTINE VECTORS

~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~

(1) DSRLNK EQU >FFD4

--- -----------------

This provides a bare-bones DSRLNK function equivalent to that in the TI-Writer Editor but corrected to be compatible with multiple RS232 cards. It is for file access only and does NOT take a following DATA 8 or DATA >A instruction. This seemed like a good idea in pre-historic times when FUNNELWEB was only a TI-Writer interface and has stuck for reasons of consistency ever since. Never mind, this sort of thing happens all the time - MS-DOS still smells like CP/M and Intel's i486 microprocessor still suffers gigantic hangovers fron the 8080.

After the usual setup of a PAB in VDP and PAB pointer in PAD, call the DSRLNK as

```
        BLWP DSRLNK
        JEQ  error handler
```

Further error trapping from the PAB error bits and GPL status is up to the programmer. FUNNELWEB utilities regularly use this DSRLNK for subprogram access by the code sequence

```
    DSRTYP EQU  >7A
```
~~~

```
           MOV  DSRLNK+2,R11
           INCT DSRTYP(R11)
           BLWP DSRLNK
           JEQ  DSERR

           DECT DSRTYP(R11)
           success exit

    DSERR  DECT DSRTYP(R11)
           failure exit
```

The offset value DSRTYP will be maintained at this value. All exits must restore the offset value correctly.

```
 (2)  KSCANA EQU  >FFD0
 ---  -----------------
```

This is an enhanced KSCAN routine which may be used in place of the normal E/A utility. It provides some extra functions transparently by setting processor status bits, which may then be tested by various conditional Jump instructions as required. There are 6 condition bits available, and the tests check these separately.

```
           BLWP KSCANA
           JNC -->> <escape> --> Jump
           JOP -->> <No Key> --> Jump
           JNO -->> <Proc'd> --> Jump
           JEQ -->> <OldKey> --> Jump
           JGT -->> <Enter>  --> Jump
           JH  -->> <Redo>   --> Jump (New key only)
```

<Escape> senses both <fctn-9> and <ctrl-C>, and <proceed> checks both <fctn-6> and <ctrl-A>. Finally an unconditional JMP may be used to loop on the keyscan after one or more other tests.

```
 (3)  FILENT EQU  >FFCC
 ---  -----------------
```

This provides a single line Editor typically used for entry of file and device names, and may be used in text mode as well from Vn 4.40 onwards. The flashing cursor has auto-repeat with acceleration, and the timing parameters are set on entry to FUNNELWEB. Editing keys are as described for filename entry. FILENT waits on entry until the previous key is released. Typical calling sequence is

```
           BLWP FILENT
           DATA <scrpos>
           BYTE <init offset>
           BYTE <length>
```

The initial offset gives the initial position of the cursor in the entry window, with null being the first position. The CPU buffer is assigned after entry to the routine, immediately following the workspace. The workspace is at >8302 in the PAD area, so the maximum buffer length can extend at most to the end of the DSR transient area. A calling sequence that allows another workspace/buffer area to be assigned is

```
        BLWP FWFILN
        DATA <scrpos>
        DATA <ofs,len>

 FILNT  EQU  $
        MOV  FILENT+2,R11  FILENT code pointer in R11
        RT                  Return via FILENT RTWP

 FWFILN DATA FEWSP,FILNT     User "FILENT" BLWP vector
 FEWSP  BSS  >20             New workspace
 FILBFR BSS  <as needed>     Follows new workspace
```

This is a lot simpler than writing your own routine, and may be necessary if you are already using the PAD area. Several auxiliary flag locations are associated with use of FILENT.

```
     LOWCAS EQU >FF22    Lower case allowed
     HEXDIG EQU >FF24    HEX digits only
     ESFLAG EQU >FF60    <escape> flag/pointer
```

When LOWCAS is null, the normal condition, all entries are converted to upper case as in the Editor SF/LF. FILENT always resets LOWCAS to null on exit. SETO of LOWCAS before calling FILENT allows lower case alpha characters to slip through. If HEXDIG is set before entry only valid hex digits can be entered and deletes produce "0" instead of blanks. Write a valid hex entry on the screen first as this is not checked.

If ESFLAG is null then the <escape> key is ignored internally and the key value passed out. SETO of ESFLAG is reserved for internal purposes. Any other value is treated as an <escape> address pointer. First instruction in the <esc> routine should be a LWPI as the <esc> sensing occurs in the internal CURSOR (see later) routine and will exit with the FILENT registers set.

(4) DELSPR EQU >FFC8

This one just writes byte >D0 to VDP memory at >300 to shut off the sprite list in Graphics mode I with normal E/A table positions. It is useful in quick cleanups after switching back from Text mode. Just call as is.

(5) VMBWD EQU >FFBA

VMBWD gives a VMBW function from in-line data, which saves program space when using fixed value data. Null length is just ignored. Call as

```
        BLWP VMBWD
        DATA <scrpos>
        DATA <CPU data>
        DATA <length>
```

The primary FUNNELWEB workspace is at >FF7C, and R9 of this workspace always points to VMBWD so that it may be called from this workspace as BLWP *R9 making a 4 word total call.

```
 (6)  VMBRD  EQU  >FFB6
 ---  -----------------
```

This is the VDP read version and is essentially similar to VMBWD.

```
 (7)  VSBRD  EQU  >FFB2
 ---  -----------------
```

The calling sequence for this is

```
        BLWP VSBRD
        DATA <scrpos>
```

and it returns the byte in VDP memory at <scrpos> to the MSB of R0 (not the usual R1 in E/A code).

```
 (8)  VFILL  EQU  >FFAE
 ---  -----------------
```

This fills a block of VDP RAM with the byte value in the MSB of R0. The FUNNELWEB main workspace FWREGS EQU >FF7C maintains R8 as a pointer to this so that it may be called as BLWP *R8 etc. Standard calling sequence is

```
       LI   R0,<MSB value>
       BLWP VFILL
       DATA <scrpos>
       DATA <length>
```

```
 (9)  VMBWR  EQU  >FED4
 (10) VMBRR  EQU  >FED0
 (11) VSBWR  EQU  >FECC
 (12) VSBRR  EQU  >FEC8
 ---- -----------------
```

This bunch of VDP utilities is equivalent to the standard E/A utility set in usage. The VM routines check for and ignore null length values in R2.

```
 (13) VSTRW  EQU  >FEC4
 ---- -----------------
```

This writes the body of a string with leading length byte to VDP. Call as

```
       LI   R0,<string ptr>
       BLWP VSTRW
       DATA <scrpos>
```

If the length byte of the string is null then the call is ignored.

```
 (14) CURSOR EQU  >FEC0
 ---- -----------------
```

The CURSOR routine is called internally by FILENT, but may also be called externally. The calling sequence is

```
       LI   R6,<scrpos>
       BLWP CURSOR
```

CURSOR flashes at the screen position specified in the calling R6, and returns the key-value in the MSB of the calling R2, and this is left in raw state at console KEYRT (>8375) . See FILENT for a discussion of <escape>. Normally CURSOR provides the up-case function for FILENT, so the LOWCAS flag is effective here also. It is not cleared however except by the <escape> path. If you want to use CURSOR externally with hex digit validation, you will need to supply your own validation routine. A table of hex digits "01..EF" is provided at

```
     HXTAB  EQU  >FEA4
```

The cursor flash rate and auto-delays are slaved indirectly to the VDP vertical interrupt rate. On entry to FUNNELWEB the number of console keyscans that occur in an interval of 14 vertical interrupts is counted and stored at

```
     REPETS EQU  >FEE4
```

This does not account for the difference between PAL and NTSC consoles, but this will be minor compared to possible Geneve to 99/4a difference.

```
 (14) DSRREN EQU  >FEBC
 ---- -----------------
```

DSRREN is intended to give a direct DSR re-entry from saved values, as is done in the E/A object loader, or more comprehensively in LINEHUNTER. It re-enters the last DSR ROM at the same entry point directly without having to search for it. All PAB information necessary after this DSR entry must be supplied as normal. It is used in DiskReview (80-col) as the new speedup for Vn 4.40 of the Myart file viewer. It is of course not necessary to set the DSRTYP again for sub-program (eg sector read) re-entry as this is only needed to find the entry point in the original search.

The relevant DSR values are stored in a 4 word block starting at

```
     SAVENT EQU >FF46
```

on every full DSRLNK call.

(15) QDCODE EQU >FEB8

This is mostly of use internally after the AID key has been detected in the path with SETO of ESFLAG (>FF60). It checks the value at AIDFL (>FF3A) and if null returns directly, else it branches to the address pointed to by AIDFL. Usually this is a QD test/load/branch routine.

(16) SETGRD EQU >FEB4

This sets the GROM address using the system GROM pointer in GPL R13, so that Module Library banking is supported. Call as

```
      LI   R0,<grom address>
      BLWP SETGRD
```

(B)   INDIRECT BL ROUTINE POINTERS

All of these routines are conveniently called by

```
      MOV  BLPTR,R11
      BL   *R11
```

together with whatever register and/or in-line data is appropriate for the particular BL call.

(1) CFILE# EQU >FFA6   Uses R0,R1

This takes a trailing DATA item >0n00 where "n" is the number of file buffers to be set aside in VDP as would be done with CALL FILES(n) from BASIC. The routine is compatible with V-9938 based systems, and rewrites MAXMEM (>8370) to suit. A new VDP header is written for TI/CorComp disk controllers and VDP cleared above that to the top of the disk DSR area. No range checking is done on "n".

(2) KEY   EQU >FFA4   Uses R2

This keyloop enables interrupts, follows the previous prescription for <escape> and ESFLAG (>FF60), and calls QDCODE if AID is pressed. The key return is placed in the MSB of R2 and left unchanged at KEYRT (>8375).

```
 (3)  RESTR  EQU  >FFA2        Uses  R0.R1,R15
 ---  -----------------
```

This does various housekeeping tasks to set up the E/A Graphics I mode for the FUNNELWEB Central Menu screen. It resets the VDP registers to E/A default, rewrites the color table, clears the screen, calls DELSPR, restores the CHR(>81) pattern to underline, and restores the key-unit to 5.

```
 (4)  RDDEV  EQU  >FFA0        Uses R0,R1,R2,R3
 ---  -----------------
```

The RDDEV routine handles the details after screen entry of a file or device name, saving the device name to CPU RAM, and building a PAB in VDP for DSRLNK. RDDEV is called externally as

```
       MOV  RDDEV,R11       Get pointer
       BL   *R11             Go to routine
       DATA <screen ptr>     Entry pointer on screen
       DATA <cpu ram bfr>    CPU RAM buffer for name
       DATA <vdp PAB adr>    VDP address for PAB
       DATA <cpu PAB data>   PAB data storage address
```

The first data item is the VDP pointer for the filename as entered on screen. RDDEV parses this up to the first space and enters it with leading length byte at the CPU RAM buffer pointed to by the second data item. It then builds a PAB in VDP at the address in the third data item, using PAB data in CPU RAM pointed to by the fourth data item. The CPU RAM buffer may well be a continuation of the PAB data, but need not be. While at it RDDEV clears the GPL status byte (>837C) and loads SCNAME (>8356) for DSRLNK.

```
 (5)  QDLOAD  EQU  >FF3C    Uses R0,R1,R2,R10,R12
 ---  ------------------
```

System files with 2 character filenames in E/A program file format may be loaded without executing by QDLOAD. A typical calling sequence is

```
       MOV   QDLOAD,R11    Get pointer
       BL    *R11            Execute
       DATA  'UL'            2-char filename
       DATA  ULPOS           Load position in CPU
       DATA  ULLEN           Load length
       JMP   <error return>
```

Successful loads step over the error branch. The normal E/A header is ignored in favor of the load target position and length. The maximum file length allowed is >1100 which is long enough for QD or QF. The file is loaded from the FUNNELWEB boot path as for other system files.

```
 (4)  FILCLN  EQU  >FEDC    Uses R0,R1,R10,R14
 ---  ------------------
```

This routine cleans up the mailbox (>A000 - >A050) area, supplying the Utility pathname if nothing starting like a valid filename is found.

(C) DATA POINTERS
~~~ ~~~~~~~~~~~~~

Other BL routines, data, code fragments, and pointers are for internal system use and may be subject to change in function or auxiliary conditions. Some not already mentioned which may be useful are

```
     NAMBUF EQU  >FF62
```

The space from NAMBUF to FWREGS (>FF7C) is used a filename buffer by the various loaders. It is long enough to accomodate length byte and 25 character name (DSK.volname.filename or whatever else fits). Increase in this was the one change that would have forced major incompatibilities with previous versions, and so it has been left as is for better or for worse.

```
     BTLN   EQU  >FF58
```

This points to the length byte of the boot pathname and all references to system filenames are by offsets from this value. The Editor printer device name, as a survival from much earlier versions, though part of this block has its own pointer at EDPRNT EQU >FF1E. All these path/file and device names are stored with leading length byte, unlike the workfile name in the mailbox at >A000 which is stored without length byte as required by the Editor.

```
     INCOL  EQU  >FF26
```

The word value here is an index, range 0-9, into the 10 byte table of color bytes for use in VReg #7, and pointed to by COLRS EQU >FF1C. Standard Funnelweb system practice is always to use this table as the source of color combinations, and to use INCOL as index. This helps avoid unexpected and jarring color changes on return.

```
     CPUDEL EQU  >FEE2
```

The value here estimates CPU speed as seen from normal expansion RAM by counting a loop against the VDP interrupt timer. It is handy as a value for CPU delay timing of beeps and bloops. Use REPETS (see earlier) for keyloop timings.
~~~

FUNNELWEB Vn 4.40 BUGS PROBLEMS

(1) Bugs and Mods

The bug report is being zeroed out at the time of first issue of Vn 4.40 of FUNNELWEB, but new bugs are being created all the while. That's the price of progress! Fixes and changes since that release are listed in chronological order below. Minor changes in associated document files may well be made without explicit mention, or other non-functional changes.

Bug ..... After use of Myart file View from DR80, the XB RUN from DR80 engages XB but then locks. <Quit> is still there though. The cause is not yet obvious.

System bug ..... Anomalous error recovery behavior may occur with SD and the DRs in systems with 80-track Myarc FDCs. See FWDOC/DR41 or /DR81 for details.

May/30/91 ... First issue of Vn 4.40.

Jul/26/91 ... Fixes for LOAD's XB user list program loading, and for ED80 <P>Dir function from SD. FILENT routine in FW/LOAD now clears the last char in the window with <fn-3>. ML40 repaired.

Oct/30/91 ... Back to programming with renewed vision !! Read/Write failure indicators added for DR file copies. DSKU notes handling revised and corrected. FWDOC files /DR40,1, /DR80,2, /PSRV updated. Dual 80/40 Editor, files ED80 ED81, now windows correctly in 40 col mode. Cause of error recovery bug in DR narrowed to Myarc 80-Tk FDCs.

If you come across a later version of FUNNELWEB, use it, but it is a good idea to update ALL files. Significant changes are indicated in the update notices. Use CF/CG and your SYSCON file to re-customize the LOAD and FW files if these have been updated.

(2) Programmers' reference

The FUNNELWEB program has to interact with a number of external programs which (to TI's eternal shame) were never intended to work together, so its internal structure is an ad hoc response to many independent pressures. As the scope of FUNNELWEB grew it became necessary to define an interface so that other parts of the program could call on routines or data in the main body of the program. This has grown somewhat haphazardly, but at this stage in the history of the 99/4a (and level of fairware support) it is unlikely to be reorganized. Details which may be relied on by external programmers are given in the supplemaentary file FWDOC/PSRV. A set of articles "Living with Spiders" appeared in the HV99 Newsletter and on some BBSs, which goes into coding details, and is still relevant.

The FUNNELWEB LOAD program is composed of 4 parts, the Extended Basic code, the XB user list data, and 2 pieces of machine code tucked between the XB code and the top of memory. DSRLNK (TI-Forth form - you handle the errors) and GPLLNK are in a >FA byte block immediately above the XB program code. These are shifted to follow the normal XB Utilities and the memory pointer updated, and are available for use by XB programs loaded from the XB User List (BLWP at >24F4 and >24F8 respectively) provided CALL INIT or CALL ILR are not invoked. This is followed by code and data used only while XB is still preserved. This same area is used later by UL, ML, FSAVE, and the ScriptLoad error handler. When FW/UTIL1 is prepared the C2 character set and FW/UTIL1's entry code are tucked between the entry point (>E006) and the actual FUNNELWEB code to speed up loading.

The end of the program is kept fixed at >FFD7 so the interface block items are at known addresses. The >FFD7 limit is compatible with E/A and XOP 1 which is not used by FUNNELWEB (but it does make transient use of the area above >FFD8, without altering the Load Interrupt vector at >FFFC).

(3) GPLLNKs with FUNNELWEB

From Vn 4.40 onwards GROM address setting from within the program and DiskReview calls a reference block routine which uses GPL workspace R13 as base value (see FWDOC/PSRV). Whenever the central menu screen is displayed the following conditions have been established.

(a) The GROM address is left pointing to an XML instruction in cartridge GROM, else in console GROM 0, and this address is also saved in the program.

(b) The GPL stack is dropped to just one entry which is this value copied to >8380.

(c) The table address pointed to by the XML instruction is loaded with the FUNNELWEB re-entry address. For maximal compatibility with existing software the XB return is at >2000 and the E/A return is at >2002. All others use PAD (>8300).

With these preparations it was possible to write a GPLLNK to fit invisibly in the E/A utilities without altering the FFALM pointer. In any event the E/A GPLLNK cannot be used because it branches to E/A module GROM code. Programs run other than from E/A or XB under FUNNELWEB may give trouble if not written to preserve the contents of location >8300.

(4) Returns

The easiest returns to FUNNELWEB are when FUNNELWEB is still all there unmolested, from >EBC8 to >FFD7. FUNNELWEB always hands over with R11 loaded with the return address. Any of the standard returns to GPL will return to FUNNELWEB provided that the XML table entry is preserved for B at >70 returns.

If FUNNELWEB is overwritten the simplest return is to the Title Screen. If your program doesn't make extensive use of VDP memory, it may be possible to stash FUNNELWEB (from >E9B0 to cover UL or else from >EBC8 to >FFD7) there below the disk DSR buffers and to restore it to CPU RAM after an irrevocable decision to exit has been made, and then a normal exit as if to GPL may be used.

If memory usage is too complete to allow this, FUNNELWEB may be reloaded from disk as the program file FW or UTIL1. Only a very simple loader is needed as there is only a single file and the memory location and start address, (>E006) and the length (through >FFD7) are prior knowledge. Strictly, only Atrax Robustus can get away with not deriving the start address and length from the file header, but given the present level of fairware support and future of the 99/4a system, it is unlikely there will ever be enough new developments to change this. It is recommended that the loader first try to load the file under name FW and if this is not found then to try it as UTIL1 before taking error action. If the application program is such that disks may have been swapped around, prompts should be issued for the primary and secondary disk numbers or the original drive or pathname assumed, as appropriate. The flag to turn on boot disk tracking in FW is at BTFLAG EQU >FF1A. A null word here disables the tracking, in which case primary and secondary system drive numbers in ASCII form must be provided in the bytes at RDISK EQU >FF18 and >FF19. Rewrite the drive numbers after loading but before handing over if these have been changed. It is always necessary to suppress boot disk tracking when split T-W and E/A access is needed.

The flag for a hard pathname is at HDFLAG EQU >FF02 and the pathname is at the address pointed to by the value at >FF58. The length byte includes the standard 2-character system filename length as well. This flag should be checked first and used by program reloaders before checking the other paths. The DM-1000 supplied in earlier issues followed this prescription.

Programs intended to work only with FUNNELWEB may re-enter it at the Central Menu Screen by LWPI of the FWB main workspace at FWREGS EQU >FF7C, SETO of R13, and CLR or SETO of R4, followed by a branch to the address contained in location CMSRET EQU >FF5C. Do not alter R8 and R9 of this workspace. A program (Opt 1-3) can tell that it was loaded from FUNNELWEB by comparing R11 at entry with the word at LDR11 EQU >FF9C.

(5) Filename transfers

FUNNELWEB has a standard way of transferring the DV/80 workfile name from one utility to another. The file name is stored on final exit from a component program at MLBOX EQU >A000 without any length byte, and padded with spaces out to >A050. This is the "mailbox". A file name so stored will survive OLD or RUN of an XB program or passage through the Title Screen. The utility entry routines inspect this location for the 2 ASCII characters "DS" or "RD" or "WD" or "HD" and if found the block as far as the first space is written to the appropriate buffer in the utility program, and the rest of the block is filled with blanks.

If utility programs are not to destroy the resident file-name then they should not overwrite this area, or else should restore it before exit, or load the file name to be passed back. Test with Formatter and Editor to see if this has been done properly. Modification of existing programs not available as source code may require a little ingenuity, and is not always possible.

(6) Standard Utility Loading

In contrast to versions of FUNNELWEB earlier than 4.30, the loads of the Editor, Assembler, and Formatter files are now just normal program loads, Option 1 for the Editors and Option 2 for Formatter and Assembler. This means that other files may be substituted for these if desired, as long as the same leading name is used, eg FO for Formatter. Test and/or correct for proper returns (see (4)) using one of the configurable menu slots first.

(7) Problems

Various problems have been observed by users of FUNNELWEB. Many of these are intrinsic to the programs being loaded, and these we can't do anything about. If you have commercial programs which you would like to use with FUNNELWEB but can't for one reason or another, particularly because of protection schemes, you should contact the sellers or originators of these programs to make your problem known. Good luck Charlie Brown! Other problems are associated directly with use of FUNNELWEB.

(a) Loading difficulties

Lockup during file loads occurs because some part of FUNNELWEB has been overwritten. The code for program file loads is located close to the top of high memory and will be destroyed if overwritten by a file that is not the last one to be loaded in normal E/A sequence with the last byte incremented for each successor file. Altering the file sequence to put the overwriting file last in a multiple load will usually solve the problem. Use DiskReview to correct the load flag in the first word of the files being interchanged (>0000 for the last file, >FFFF for all others) and DiskReview or other DM program to interchange file names. File lengths of >2000 in 33 sector files, such as produced incorrectly by E/A SAVE, may need to be corrected to >1FFA if the changed order would result in overwriting of previously loaded code.

Load/Run of object files is handled by the Loader in the E/A utilities. Autostarting of files is by a direct branch from the Loader to the program code, so that a autostarting object file may overwrite FUNNELWEB. Files that are too long may refuse to load. Unsuitable absolute file addresses may destroy the code for DEF table entry.

(b) Running difficulties

The program file loader gives a fair approximation to GPL or E/A module environment. If problems are observed they are usually associated with key unit choice and key response in the program loaded, particularly with programs that set a key-unit of 0 instead of choosing the one they really want. The cartridge loaders now reset the GPL stack pointer at >8373 to >7E which seems to help.

Programs that sense which module they are running with and adjust their action accordingly can cause problems. Since the point of FUNNELWEB is to make all sorts of programs work with modules they weren't intended for or even no module at all, it has already taken the necessary actions and must not be second-guessed by programs it has loaded. The only solution is to modify the programs or else not use FUNNELWEB. Running from the E/A module gives the easiest way to switch back and forth between FUNNELWEB and a standard environment.

Some programs in E/A program format contain code to unload the standard utilities from the E/A GROM because the E/A module does not do this for Run Program File. The c99 memory image files prepared with the original unmodified C99PFI use this method to force the utility load. Running or return from such programs may not be graceful if FUNNELWEB was loaded from E/A. c99 is smart enough not to try getting code from non-E/A GROMs, but other programs may need revision.

FUNNELWEB has also adjusted the E/A XML object file loader path for its own purposes. If an assembly program needs to load an object file it should use a direct BLWP at LOADER to the E/A routine. Error returns may need special attention.

(c) Exit difficulties

Read (4) above on returns. Opt. 2 loads will be necessary for program files that overwrite the E/A utility area in low-mem but still use a GPL return via the contents of R11 as would be acceptable for a Opt. 2 (or E/A 5) program file load.

Some programs which do not appear from their load size to overwrite FUNNELWEB may in fact corrupt it while running. Some particular cases of pre-existing programs may be helped with minor changes, as a return to the title screen is usually preferable to a lockup.

An example of this kind is the Dragonslayer Spellchecker. An early version inspected of this commercial program runs normally under FUNNELWEB with load parameter 2, but crashes the machine on exit. This is improved by changing the words 0460 0070 to 0420 0000 to return to the title screen. Use HEX string search in DiskReview to find this exit.

Another example needing modifications to exit code is DSKU which leaves interrupts on, the interrupt hook loaded and pointing to an address overwritten by the incoming FW.

FUNNELWEB Vn 4.40 LOADERS

The FUNNELWEB system includes special auxiliary loader files LL, SL, and User List files UL and ML, which greatly enhance the normal E/A module functions.

LOW MEMORY LOADER LL

Some well known utilities for the TI-99/4a, such as the Editor and Assembler occupy low memory and use high memory for extra code, but mostly as a single large data storage area. The only way that TI provided for users to load such files in object format was the Minimem module, and now Low-loader provides this function for general use, with automatic recognition by the FSAVE program (see FWDOC/UTIL).

When LOW-LOADER is selected from the Load Environment screen, the LL file is loaded from the boot disk/directory. This provides an alternate set of E/A utilities just below the FUNNELWEB program in high memory. Low memory is now used as the first block for loading relocatable object files (only 8K is available in this block and 17K in the high block). All E/A REFs are recognized, but E/A program utilities REFed this way will not be available to program file loads of FSAVEd versions. The predefined REF/DEF table starts at >E200 and ends at >E138, and new entries build down from there.

If low memory is used from >2000 up then the normal E/A (>2002) or XB XML (>2000) target locations are no longer available, so LL sets PAD (>8300) as the XML address for GPL returns by selecting a >F0 XML in console GROM.

SCRIPT FILE MAKE/LOADER SL

In Vn 4.40 of FUNNELWEB the functions of Script-Load have been extended to include a full Assembler MAKE facility as well as multi-file object loader scripting. SL recognizes a list of up to 15 object files specified in a DV/80 script file. Just how it handles this list depends on how the auxiliary loader file SL is invoked.

If Script-Load is called as an loader option as configured in the Central Menus or in User Lists, or else manually from the DiskReview file loader, it will fetch the nominated Script file, read the object files to be loaded, and proceed to load them. Any set of relocatable object files loadable by E/A will load without restriction. The normal caveats on absolute files still apply, but with an additional restriction that AORGed code may not be loaded into PAD or above >FFD7 (only silly protection schemes ever did this anyway). The virtue of the Scripted load approach is that it removes the tedium of entering a whole set of object file names, and also allows the RUN link-name to be activated automatically.

When SL is loaded as Option 5 SCRIPTLOADER it will prompt for entry of a Script file name, but after this has been entered some further options are available. The first query is

```
MAKE Assembly ? N
```

with N as default for immediate <enter>. If the default is chosen the object load function proceeds as from external access. If however <Y> is pressed before <enter> the Assembly MAKE is invoked and a further query asks if the Loader should be automatically entered after successful assembly. The same Script file is used for both MAKE and LOAD functions. The MAKE function is especially intended to reduce the tedium in SAVEing program files in complex assembler or c99 projects. If an error occurs during assembly you are returned to the FUNNELWEB main menu.

File SCRIPT included as an example in the FUNNELWEB Vn 4.40 package is the actual Script file used for complete re-assembly and program file preparation for the 80-col DiskReview. A full discussion of the details of Script construction follows.

A Script file is prepared as a normal Dis/Var 80 text file with the Programmer's Editor, or any other way. The overall structure is reminiscent of an assembly source file, except that there are no labels. Your instructions are carried by directives, which may be followed on the same line by data such as file names where appropriate. Comment lines in a Script start with an asterisk in the first column and blank lines are ignored. Comments may also follow entries except as noted. This largely follows TI Assembler conventions.

Details of the MAKE and/or LOAD process are controlled by these directives, which are all pre-defined words of 4 upper-case characters. These fall into several classes, some illustrated in SCRIPT. All files specified must be Dis/Fix 80 object files. At present no control is provided in MAKE for LIST device entry or for assembly options other than the FUNNELWEB default of RC. These can be added if demand is apparent.

(a) File specification

FILE "DSKn.xx" .. Followed by a filename complete with pathname (up to the maximum length allowed for a floppy disk name), all in quotes, single or double, FILE specifies one of the files to be loaded. No spaces are allowed between the quotes.

BOOT "filename" .. Followed by a file or child path name without any load pathname. This instructs the ScriptLoader to supply the FUNNELWEB boot pathname and to append this file/path name. If the Hard Disk path is OFF the drive number where SL was found is used as "x" in "DSKx.filename".

UTIL "filename" .. As for BOOT except that the currently defined Utility pathname is used.

(b) Loader control

AUTO .. If AUTOmatic running is specified the link name specified after the LAST directive will be used to autostart the programs. The default, with no AUTO or ALLM directive, stops for editing of the link name.

IAOF .. I(nternal) A(utostart) OFf cancels internal autostarting of object files (like Option 8).

LWLD .. LoWLoaDer sets up the load conditions as for Low-Loader (Option 6 of the Loaders screen). File LL is loaded from the boot disk after the Script file has been read but before object file loading begins. It should be issued at the start of a Script and over-rides ALLM.

LAST .. Indicates the end of the script to be parsed by SL. If it is followed on its line by text, the first 6 characters will be read as a link name for RUNning the programs. If the end of the script is reached before a LAST directive is found, an error is issued. This allows a single DV/80 file to do double duty as a full document file and a ScriptLoad file.

(c) Memory Control

ALLM .. ALL M(emory) sets the LFHM memory pointer to the standard E/A value >FFD7. Once set it cannot be revoked. It also sets AUTO and a link name must be specified on LAST or an error will be called. This is because the FUNNELWEB object loader code can no longer be assumed to be uncorrupted. Returns are adjusted to be to the title screen for this same reason.

(d) Assembler MAKE directives

ASSM .. Object file names following this directive are checked to see if they end with /O (or ;O for c99 users), in which case the last character is changed to S and the name displayed for passing to the Assembler in its source file version. The Assembler will produce an object file in the form specified, following the normal FUNNELWEB convention. If a object file name does not end this way, it will be displayed with a preceding asterisk but not re-assembled. ASSM is the default condition for the MAKE pass.

STOP .. Object filenames following this directive are totally ignored by MAKE. ASSM after STOP gets attention by MAKE again, and these directives give complete control over the files to be assembled. Loader and Memory control directives are ignored by MAKE, and the LOAD pass ignores ASSM and STOP.

Error handling in the Load process is now as complete as for the other object loaders, with additional features. When an Undefined REFerence error occurs, up to 20 undefined REFs are displayed. Also <enter> from an error display will execute a cartridge ROM program if available - very convenient if you have DEBUG or SBUG II available there.

USER LIST FILES UL
~~~~~~~~~~~~~~~ ~~

User List files are a package of loader interface and filename data in a fully position independent executable program file. User lists are configured as part of the system configuration CF/CG operation. Unlike the XB user list which is part of the LOAD file and stored along with other data in SYSCON files, as many user list files as desired may exist in the system under different names and each is configured separately. One user list may call others allowing a tree of choices as large as your disk system will permit. Any type of file may be specified in a User List, including Script-Load files which allow object files to be fully automatic in loading and running.

The configuration process using CF/CG allows a filename only to be specified on the Boot or Utility paths, or else a a full filename in form "DSKn.filename" or general path/file name to that length. You may also call for a pause and reminder before loading. Normally entry #8 on the Word Processor side of the Central Menu is reserved for a User List file named UL. This will be a little "sticky" in that it will not always need to be reloaded from disk when reselected. User List files may be called from any Central Menu entry, and if so these should be configured as "Other" to avoid overwriting the main UL.

User Lists should be loaded only from the boot disk path under a 2- character filename, and if a load from a UL fails, you should cancel immediate reload by <esc> from the central Menu Screen.

MULTI LIST FILES ML
~~~~~~~~~~~~~~~~ ~~

FUNNELWEB Vn 4.40 takes advantage of the new FW program services to allow an expanded User List facility in the form of Multi-List files. A Multi-List program file treats a nominated set of User Lists as data objects. A ML file loads the nominated UL type files and presents their contents in vertical columns on the screen. The standard 40-column version allows 3 User Lists to be displayed at one time, and the 80-col version allows up to 6 User Lists to be displayed at once. The primary restriction is that the User Lists read in as data by any Multi-List must all be locatable on the FUNNELWEB boot disk/path. The programs called by these User Lists can be anywhere and can include new Multi-Lists, or User Lists. Choice of the program to be loaded is by cursor driven selection and <enter> or <R>un, which then transfers control to the loader code in the particular User List to work on its own data.

If <C>hange is pressed then the 2 letter User List names may be edited and the new set will then be loaded. In the 40-col version only, pressing <N>ext automatically swaps the current set of 3 User Lists for an alternative set of 3. If a name is blank, that column will be bypassed, and also if the nominated UL is not found. At present there is no provision for saving edited data, and this must be set up with DiskReview's sector editor. Call up the first sector of the Multi-List program for ASCII editing, and starting at byte >08 (after the 3 word header and the initial JMP instruction), there are six 2-character filenames to be entered. If you don't have six candidates, leave the remainder as spaces.

Either of the 40- or 80-col Multi-Lists may be called UL and substituted for the User List of that name, and any call of a Multi-List will overwrite the current Central Menu User List. Remember then to rename the existing UL and to edit the load list in the Multi-List. Unlike the User Lists, Multi-List files are not position independent and always reside at the system's UL reserved position at >E9B0 in >218 bytes or less. The limited size also accounts for their fairly spartan appearance. Multi-Lists may also be loaded from any path under any name, and always cancel automatic reload on 8 USER LIST when a loader selection is made. Calling one Multi-List as a data object (as distinct from a program to be loaded from one of the ULs) from another will give an obvious nonsense display in that column.

FUNNELWEB Vn 4.40 UTILITIES

Utility files of various origins are included on the FUNNELWEB distribution disk(s). Files LGEN/S (for generating templates for XB programs like LOAD) and LDSR/S for use with LowLoad from Vn 4.0 are not included as not a single comment has been received concerning their existence or use, but are still available on request. This file contains notes on the following programs

1 CF/CG
2 CP
3 FSAVE
4 LDFW
5 UL
6 CT8K/O
7 LH
8 XB4THLD

(1) Configuration CF

CF/CG is used to customize LOAD, UTIL1/FW and various User List files to your preference in the run-time setup of FUNNELWEB. The program makes extensive use of windowed displays and context sensitive help screens. The editing process is tree structured and is easy to follow along to any particular item. Help screens are often available with a press of "?" <fctn-I>. You save system configuration details for re-use in data files of which SYSCON is an example. CF loads as an Option 2 program file from FUNNELWEB, which is used as a source of default data but the program in memory is NOT altered. Remember to use Install before exit and RELOAD to check your handiwork as the FUNNELWEB in memory is NOT altered in the configuration process.

(2) c-Compiler interface CP

Clint Pulley's Vn 4.0 c99 compiler files C99C/D/E may be loaded directly as Option 3 program files. File CP brings user convenience in working with c99 closer to FUNNELWEB standards. Instead of loading c99 directly, load CP as a Option 3 program file, and it will then load C99C/D/E from the same disk drive. The return from c99 reloads FW from either the set pathname or the E/A side boot drive and preserves the mailbox filename. If no filename was initially present, the c99 source code filename is installed. See also FWDOC/SCLL for details of ScriptLoad which will be found very handy in assembling and/or loading the collection of Object files which usually make up a c99 program.

(3) Save utility FSAVE

The E/A SAVE utility loads as absolute code in low memory. FUNNELWEB is compatible with SAVE, but does take up its own 6K share of high memory, so the FSAVE utility has been prepared to allow SAVEing of object files loaded by FUNNELWEB, including into low memory. Refer to the E/A manual for general information.

FSAVE loads as absolute code overlaying the FUNNELWEB (UL) system area. The start and first executable instruction in your own code should be DEFed with SFIRST and the last address DEFed by SLAST. Select entry point SAVE and enter the filename to which your program is to be SAVEd in E/A compatible memory image format.

If the Loader has placed files so that SFIRST is in hi-mem and SLAST is in low memory, FSAVE will SAVE high memory from SFIRST to the FFAHM indicated by the Loader at UTLTAB+2 and then proceed to SAVE low memory from >2676 (above the E/A utilities) to the FFALM. The utilities are not included so that the files will remain compatible with FUNNELWEB if reloaded under a different module.

When used with Low-Loaded (Opt 6) files, FSAVE saves its first module from low-mem from SFIRST to the top of lo-mem, nominally >3FFA (at UTLTAB+4), and then from hi-mem from >A000 to SLAST. If SFIRST and SLAST both point to the same segment the SAVE is normal. The MBSAVE entry adjusts the hi-mem start to >A050 above the Mailbox. Use E/A SAVE for addresses in the >6000 to >8000 cartridge space.

The MEMSAV entry point allows direct entry of hex address limits for the memory block to be SAVEd. The second entry is the address of the last word (inclusive) to be saved. MEMSAV ignores SFIRST and SLAST but these must have been DEFed, perhaps by a dummy object file, for correct LOAD/RUN operation.

FSAVE indicates the actual length of the memory block saved in each file in the second word of the header block, to a maximum of >1FFA in each file. The TI E/A SAVE utility, amongst its other little foibles, adds a further 6 bytes to this count, but the program file loader in the E/A module believes the byte count in the header. In normal usage the extra 6 bytes, falsely indicated by E/A SAVE, as read in from VDP to CPU RAM do not cause any problems. FSAVE files will of course not cause any problems unless perhaps a loader incompatible with E/A is used.

File FWSAVE for cassette or long file saves has been removed from the package as no comments were ever received concerning its use. A revised version which works with Vn 4.3x exists and is available on request.

(4) Basic loader LDFW
~~~ ~~~~~~~~~~~~ ~~~~

LDFW is an auxiliary load program in the form of an autostarting relocatable object file which may be executed from E/A, Minimem Basics by CALL LOAD(".."), Myarc XBII by CALL LR(".."), E/A Load Run, or most other object file loaders such as come with Myarc or
~~~

Corcomp disk controllers. It may be kept in Minimem cartridge RAM if you follow the MM instructions for forcing it to load there. The RUN name is LDFW. It offers choice of several pathnames or entry of floppy disk numbers.

The information supplied by the user to LDFW is used only to locate the FW/UTIL1 file and does not reconfigure this file in any way. Incidentally, CALL LOAD will not work from E/A or MiniMem Basic if you have a Horizon style RAMdisk in your system with LOAD configured as a call name.

(5) User List UL

Writing in of the 8 user selectable options is done with the CF/CG installer program. If a hard pathname load is chosen CONFIG will remind you that only the file-name or further pathname should be entered. There is no entry corresponding to "9 <CRT ROM> 0" which looks for a cartridge ROM header at >6000 and "9" executes the first program listed there and "0" the second. This may be handy for owners of Vn 2.2 consoles who have disk controllers that will load FUNNELWEB.

Remember that UL is a normally executing Option 2 Program file and different UL files can be chained by specifying them in a UL type of file. UL is coded to be fully position independent. Just remember to avoid file name clashes. File D1 is a UL type file collecting various disk utilities together for Opt 3.

(6) Cartridge RAM loader CT8K/O

Object file CT8K/O is used to store FUNNELWEB in >6000 - >8000 cartridge RAM, if present, so that it comes up as a selection after the title screen (not on V2.2 consoles or in the presence of Myarc 128k OS). The code produced is ROMable. Load FUNNELWEB, make sure the character set of your choice is loaded, and then load CT8K/O as a Utility Load / Run option. FUNNELWEB is loaded into hi-mem on selection as for FW.

(7) Assembly line locator LH

LINEHUNTER is one of our working tools now made available for FUNNELWEB users, though it could well stand comparison with many commercial programs on its own. If you write substantial assembly programs you will be aware of the problems in tracking down assembly errors through multiple Copy files. Printing of List files is fine, but impractical for the home computer user. It is a dual-mode program which will also execute outside the FUNNELWEB environment.

Give LINEHUNTER the name of your master source file and a line number and it will locate and display the line itself, and the line number in, and the name of the file in which it is located.

It will conduct a similar search for a source code label if one is entered instead of a number. If <space> is pressed string search is enabled in the Operand field of valid assembler source lines, for tracking down where labels are used. The search starts automatically when 4 digits of line number or 6 characters of label have been entered, or else with the <enter> key. Pressing <ctrl-A> resumes the search process and <ctrl-C> terminates the search. Exit from the program is by the <ctrl-=> key.

Fastest searching is done when all copy directives are located in a separate short file, as LINEHUNTER does not have to examine each line in COPY files for further COPY directives.

(8) XB FORTH Loader XB4THLD
~~~ ~~~~~~~~~~~~~~~ ~~~~~~~

This program allows the standard TI FORTH disk to be loaded by TI XB. It works only with the XB and E/A modules, and its primary use would be from the XB User List.
~~~