

Documentation for

**FUNNELWEB** version 4.40

40 column routines

October 1991

Prepared by Geoff Trott

from the originals provided

by

Tony McGovern,  
215 Grinsell Street,  
Koforo, NSW 2289 Australia.



## Table of Contents

Universal Utility Environment (-READ-ME)	3
Load Information (FWDOC/LOAD)	8
DiskReview (FWDOC/DR40, FWDOC/DR41)	14
TI-Writer Loader (FWDOC/TIWR)	28
Editor Assembler Emulation (FWDOC/EASM)	31
Loaders (FWDOC/SCLL)	35
Utilities (FWDOC/UTIL)	40
Bugs and Problems (FWDOC/REPT)	44



Funnelweb version 4.40 - October 1991

Universal Utility Environment

(1) General Notes

Funnelweb version 4.40 is a powerful operating and utility system for the TI99/4A 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 (Editor Assembler or TI-Writer), 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 (Editor Assembler, TI-Writer, Disk Manager) and menu/loading facilities for a wide range of other programs. Cartridge swapping from Extended BASIC is now largely unnecessary. The basic TI99/4A 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 Mechatronic 80 column unit and the Geneve 9840 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 a 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 version 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 Editor Assembler except of course that the specific module is not necessary.

Absolutely necessary:

- (1) TI99/4A 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

4

Other items that you will need to make full use of Funnelweb version 4.40 are:

- (1) Editor Assembler, TI-Writer 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 Extended BASIC and Editor Assembler 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, 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 programmer's search utility

FSAVE - Improved Save utility for generating Editor Assembler program files

CP, C99PFIO;0 - auxiliary files for use with Clint Pulley's c99 release

CT8K/O - installs Funnelweb as menu item in Editor Assembler plus 8K RAM cartridges

LDFW - Dis/Fix 80 loader for FW from MiniMemory, Editor Assembler and Myarc XBII

XB4THLD - loader for TI-Forth from Extended BASIC 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 organization, the Extended BASIC 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 Dijiit 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 documentation for the 40 and 80 column operation with AVPC, TIM, Geneva, Mechatronics systems

FWDOC/DR80, FWDOC/DR81 and FWDOC/DR82 - documentation for 80 column DiskReview

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

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 column group contain Formatter commands, while the DOC files in the 80 column group should be printed from the Editor. In particular FWDOC/PSRV contains assembly source code examples containing "at" signs.

(3) Update Notes

Version 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 version 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 version 4.30. They still can be used with version 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 column 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.
- (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 FWD0C/PSRV and FWD0C/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 Extended BASIC module in place and to create a system which makes the most of the strengths of the TI99/4A system, without straining to imitate any other operating system. 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 Editor Assembler plus RAM cartridges and the Dijit AVPC card. The backup system has Myarc 512K RAMdisk and Mechatronic 80 column unit. Specific extra program capabilities or bug work-arounds have been made, in the flying blind category, for the Geneve 9840 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 TI99/4A. 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 version 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 to \$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 money received is 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 cannot afford the time for, or cost of routine mailing of updates and do not 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 TI99/4A are stretched. Our normal 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.

October 30th / 1991  
Tony and Will McGovern  
215 Grinsell Street,  
Kotara, NSW 2289  
Australia



Funnelweb version 4.40 Jul/26/1991

### 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 Editor Assembler 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 Editor Assembler option 5 and may be renamed UTIL1 for easy loading as DSK1.UTIL1 from Editor Assembler 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 version 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. <esc> as <FCTN-9> or <CTRL-C> will bring up and hold a second screen.

With Extended BASIC 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 Extended BASIC but not with Myarc XB-II).

Object file LDFW auto-runs from any compressed object file loader. CT8K/D 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 path name of the form WDS1.FWB, from a subdirectory on the WDS1 hard disk containing all the central 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 path name also. Other path names 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 Extended BASIC, Editor Assembler or TI-Writer modules.

### (4) Boot Disk Tracking

Funnelweb either from Extended BASIC or as FW (under any name in form DSKn.yyy), with hard path name 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 path name 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 Editor Assembler system utility files, as may be done in systems where a single SSSD or DSSD capacity is insufficient. CF/CQ 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 path name option and this over-rides the other choices.

The secondary system drive number is for the Editor Assembler 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) Extended BASIC Selection screens

After auto-loading from Extended BASIC, 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 Editor Assembler each lead to the Central Menu screen with the corresponding mode set. When these paths are taken the Extended BASIC environment is destroyed and cannot be restored except by rebooting Extended BASIC, which takes just 3 easy keystrokes, or from DiskReview which can restart TI Extended BASIC.

### (7) Extended BASIC 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, Extended BASIC RUN of another Extended BASIC program, or just return to the Extended BASIC command mode. LOAD may be reRUN from the command line to allow recovery from Extended BASIC RUN errors without complete reload from disk. This path should work with any dialect of Extended BASIC. Some entries have been predefined by way of convenience and example.

At this stage the set of Extended BASIC INIT utilities is still available for either Extended BASIC 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 colours

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

#### (II) Quick Directory

The AID key, <FCTN-7>, will call up from file QD a paged (by single B/N or <CTRL-E>/<CTRL-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 track disks as though they were on normal 40 track disks.

## (iii) QUIT &lt;FCTN=&gt;

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/&lt;FCTN-9&gt;/&lt;CTRL-C&gt;

In Funnelweb itself <CTRL-C> may be used instead of <FCTN-9> (BACK) as the <esc> key just as in the Editor and <CTRL-A> usually substitutes for <FCTN-8> (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 path name 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 to 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 Option 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 FWD0C/SCLL for a fuller discussion of User Lists and Multi Lists and also the CF/CG instructions in FWD0C/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 to 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 Editor Assembler 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 and FWDOC/DR41 or FWDOC/DR80, FWDOC/DR81 and FWDOC/DR82.

14

Funnelweb Version 4.40 - October 1991

DiskReview, 40 column version

(1) Introduction

DiskReview (program files DR/DS or as DR40/DR41) integrates with the Funnelweb Version 4.40 system an extensive set of disk and 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 TI99/4A.

(2) Loading

It is installed permanently as Option #8 on the Funnelweb Editor Assembler Central Menu screen as DiskReview, filename DR, with an Option 1 (Text-Mode TI-Writer) 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 Editor Assembler option 5. Screen colours 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 path name.

(4) Directory

When DiskReview first runs, it shows a screen with an information block, command reminders and a 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 sub-programs which require a unit number entered in the form DSKx., with the character "x" being in the range 1 to 9. If your choice cannot be found then the error trapping will tell you. When a 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 work file 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 DiskReview 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 DiskReview). All key entries are automatically converted to upper case except where lower case might be needed as in ASCII search string entry.

<i> to <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, DiskReview 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.

<O> key. As in Funnelweb, this cycles the screen colour choices. Also available on other directory screens.

<D> key. This brings up a menu of disk oriented functions in the right box. See later 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-8> 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> or <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>/<FCTN-X> keys. These move the cursor marker up and down the directory, paging it as necessary.

<CTRL-E>/<CTRL-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 appropriate in one or more of the block of filenames in the lower right block.

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

<T> and <CTRL-T> keys. Tag 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. Untag the file under cursor. Untag all files with <CTRL-U>.

<CTRL-A> key. Several Actions on All tagged files are available from a new command set which appears in the key advice block. Details are in the next section.

<R> key. Run program sends the marked filename to the appropriate Funnelweb loader or to an internal Extended BASIC 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 sub-programs which must be supported by the disk or RAMdisk DSR to TI99/4A 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 later for details.

<P> key. Print directory 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.

#### (8) Tagged File Operations

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

<C> key. Copies 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. The error messages indicate read or write failure.

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

<D> key. Deletes 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 Disk Utilities 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 this document for technical details and how to use the sector editor to create these.

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

o Program files

If the program file has been identified as Editor Assembler compatible, it will be passed to the Funnelweb Loaders with choices presented as for Options 1 to 3, of which 2 GPL is the normal selection, corresponding to Editor Assembler option 5. See FWDOC/EASM for details. If the program file is not a recognised Editor Assembler 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 or 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 48 sectors.

The file display in 40 columns 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-memory 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-memory 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-memory 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>/<CTRL-A> keys. Pressing <CTRL-V> or <CTRL-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> key. 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> to <CTRL-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> to <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 path name.

## Part 2 - Disk and 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 wrap around 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 track 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 sub-program is used to format the disk. Myarc floppy disk controllers with the original 40-track DSR ROM will format 18 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-track 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 number, the current disk name 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, 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 number is entered, the disk volume name is read and shown with cautionary messages. If you <PROC'D> 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 sub-program access. The only absolute address used is for reading the current EMDK for display and nothing is written to the RAMdisk 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 sub-programs are supported in the form prescribed in the Horizon/OPA Version 8.14 ROS documentation. 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 hexadecimal 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 Search 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 hexadecimal 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 hexadecimal 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 John Birdwell's Disk Utilities program in the interests of standardization.

The accelerating auto-repeating flashing cursor appears either in the hexadecimal 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 hexadecimal display accepts only hexadecimal digits for editing. A running count of cursor position in hexadecimal is displayed, along with the byte under the cursor.

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

<CTRL-S>/<CTRL-D> and <FCTN-S>/<FCTN-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>/<CTRL-Q> keys. Force the cursor and entry into hexadecimal 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 and edited sector over as the reserve copy.

<CTRL-R> key. <R>ecalls the reserve sector copy to the edit windows. The <CTRL-M>/<CTRL-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.

#### o String Search

When a string search option is selected a choice is presented for ASCII or hexadecimal string entry. Each entry mode allows specification of a wild card. For ASCII entry this has "?" as the original default and for hexadecimal search this is byte >00. When the wild-card is entered, all occurrences of 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 hexadecimal search mode allows the search to be forced to even word boundaries. This cuts down on the number of false alarms when looking for instruction opcodes or word data in Editor Assembler 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 hexadecimal 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-track 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 Extended BASIC loader, >8372 for TI Extended BASIC version 110.
- >12 — GROM address GPL stack entry for interrupt loader, normally >8303.
- >14 — GROM address for interrupt loader re-entry, normally >8495.

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 Editor Assembler Saved program files chew up 34 sectors on DSDD 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 sub-programs 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-track disks are not a common medium of exchange. Please let me know your preference.

Bug Note

A bug occurs with use of DR (and SD) on our Myarc FDC with an 80-track EPROM. 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 the FDC to enter some strange state and sensible directories are obtained only for RAMdisks at CRU >1000. 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. A fix has not yet been found.

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 78 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 >0C to >FE leaving the last byte null. If the 3 bytes from >09 to >0B 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 (>09 to >DA) and if it is not null on both source and target files, ignores the file.

October 28th / 1991  
Funnelweb Farm

Funnelweb version 4.40 - October 1991

TI-Writer Loader

If you are not already familiar with the operation of TI-Writer, some help is available from various User Group sources and the original manual may still be available. The multilingual capabilities of the version 2.0 European release of TI-Writer are not directly supported by Funnelweb, mostly because its tab records saved with files are fatally incompatible with the original.

(1) Editor

The Editor works in all respects as it does with the TI-Writer module when selected with Formatter listed on the Central Menu Screen, with the improvements listed below. It accommodates any file that will load into the original TI-Writer, with no decrease in buffer size. New facilities in the 80 column Editor designed for the Diji AVPC card and other 9938/9958 systems are described in FWDOC/EDAV in that package. The 80 column editor now supports a 40 column display mode also.

- (i) The colour selections using <CTRL-3> are the 10 configured in Funnelweb using CF/CG.
- (ii) The printer device-name is read in from the main program and used as default.
- (iii) A default Edit filename may be configured with CF/CG into Funnelweb. If left blank the default utility path name or the pre-existing filename will be set.
- (iv) The <CTN=> system Quit key remains disabled at all times while in the Editor, including SD.
- (v) SD handles disks with up to 127 filenames in a fully paged mode, not just as a one-time scroll. Paging is cursor driven or by the single <N>/<B> or <CTRL-X>/<CTRL-E> keys. Hard disk path names are not supported, so use file emulation of DSK1 for your work disk. On 80 track (DSGD) disks file sizes are shown as they would appear on a DSDD or smaller disk. The sectors Used and Available are correct however.
- (vi) An asterisk after the file length shown in the directory indicates a fractured file, one stored in more than one contiguous block on the disk. A full or nearly full disk will usually have one or more fractured files because of the way the file system allocates space.
- (vii) A Dis/Fix 80 file may be marked for the LF/SF function by pressing the <Space Bar>. The current file so marked is indicated. Pressing <O>/dF restores the workfile name existing on entry to SD.
- (viii) If <P> is pressed, SD prints the directory in two column format to the Editor print device. This may be a disk file and the device is opened in Append mode.

- (ix) If <V> is pressed, SD will "View" the DV/80 file under the cursor on the screen. Pressing any key pages through the file and <BACK> or <CTRL-c> aborts the "View". This facility is handy for quick checks of the contents of other files.
- (x) Pressing <D> for <D>eIF will delete, after checking, the file currently under the cursor. The file list and sector counts are immediately updated. DF is no longer available or needed on the command line.
- (xi) REDO causes SD to re-read the directory from the same drive. Keys <1> to <9> will cause the directory to be read from that drive, again without return to the Editor. The <D>eIF buffer is not affected.
- (xii) <U>nProtect and <S>etProtect set file protection status and the directory is immediately re-read to verify.
- (xiii) The C1 file has been altered to improve legibility. If further changes are needed, say to support a language other than English, other files may be substituted.
- (xiv) Dual tab sets are now supported and may be swapped by ST from the command line. Both tab sets are saved with the document. The original set is initially duplicated if a document with only one set is read in from disk.
- (xv) 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 little problem with loss of keystrokes and Reformat is faster.
- (xvi) Delete Lines is greatly speeded up, particularly on freshly loaded files.
- (xvii) Copy Lines is now very rapid and does not leave partial copies if 'Text Buffer Full' would occur.
- (xviii) Move Lines is now instantaneous and no longer can cause the 'Text Buffer Full' condition.
- (xix) Improved encoding has increased apparent Text Buffer capacity slightly, the increase depending on the text material.
- (xx) A right margin warning beep has been incorporated. The inset from the right margin may be set, or the beep cancelled, but it requires disk sector editing of the third word in the ED file (after the 3 word file header). It is set to 5 (0005) in the distribution version. A null word here cancels the beep.
- (xxi) Alpha case conversion is now available, 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.
- (xxii) The End-of-File message has been replaced with a full width ruler line which shifts with window and line number selection.

- (xxiii) Page up and down are now also on keys <CTRL-Q> and <CTRL-A>. Used with the <CTRL-E>, <CTRL-S>, <CTRL-D>, <CTRL-X> cursor block, <CTRL-Z>, <CTRL-W> tabbing and <CTRL-C> this gives convenient one-hand operation without the stretching needed for the <FCTN-key> equivalents. Both <CTRL-Q> and <CTRL-A> exit Command mode conveniently.
- (xxiv) The margin release key <CTRL-Y> now gives full release on both left and right margins.
- (xxv) The ED/EE files load purely as Option 1 program files. If so loaded, either of Word Processing or Program Edit mode will result depending on the details of the load path. Word Processor mode can be forced by holding down the <W> key as it finishes loading and starts to execute. Similarly Program Edit mode may be forced with the <P> key.
- (xxvi) Multi-lingual text editing is helped along in the main program (FW or LOAD) as the character file to be loaded with the editor can be selected after <escape> from the central menu screen. This presents the filename for editing, for both Text Editor and Program Edit in turn.

## (2) Formatter

The Formatter functions normally with the following exceptions.

- (i) The printer device-name is pre-configured in the main 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 path name 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 (Editor Assembler option 5) program and another Formatter program in this form may be substituted.

Funnelweb version 4.40, July 1991

Editor Assembler 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 and FWDOC/EDAV still apply. It is generally more convenient with these modifications than the Editor Assembler Editor ever was.

- (i) The Editor now functions with word-wrap disabled, Editor Assembler 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 Editor Assembler system drive. As supplied this contains a larger set than C1, quite suitable for Assembly source which typically has a high proportion of white space. 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-0> 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 Editor Assembler 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 (Editor Assembler option 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 path 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 Editor Assembler 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 to 127 loaded from GROM even if they are immediately over written by CHARA1. The utility must be in Editor Assembler 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 Editor Assembler option 5, adequate for most programs normally loaded by the Editor Assembler RUN PROGRAM FILE loader from GPL. Hand over is in the GPL workspace and the presence of the Editor Assembler utilities cannot be assumed.



(iii) Option 3 supplies the Editor Assembler RUN PROGRAM FILE function for Program files that have been prepared from Editor Assembler object files which do use the Editor Assembler utilities. Hand over is in the USRWSP at >208A, but this is not written to after file loading starts so that it does not corrupt files which ignore the Editor Assembler 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 Editor Assembler 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 FWD0C/REPT for details) and the first free address in low memory is unaltered.

(iv) To load assembly program files from cassette, use either the Editor Assembler module directly or a special loader program such as CASSLOAD from Extended BASIC.

The last program file in a sequence may over write Funnelweb at the top of high memory without trouble while loading, but if a utility over writes 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 Editor Assembler 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 Editor Assembler manual specifications should run correctly. Common sources of difficulty are discussed in FWD0C/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 Editor Assembler module). DEBUG and SAVE from your Editor Assembler package both work normally.

(i) Options 5 to 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 path name with SCRIPT as filename is supplied as default starter entry. See FWD0C/SCLL for details of SL.

(iii) Low-Loader (Option 8) allows object files to be loaded starting in low memory at >2000 and then continuing in hi-memory. All normal REFD 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 FWD0C/SCLL for details of LL.

(iv) If Option 7 is selected the LFHM is reset to the Editor Assembler default of >FFD7. This allows the last relocatable auto-starting object file to load over Funnelweb if necessary, there being no memory contention because the auto-start hands over to the program without returning to Funnelweb.

- (v) Option 8 intercepts the Auto-start 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 Editor Assembler 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 Editor Assembler 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 <FCTN-S> and <FCTN-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, <FCTN-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 version 4.40, July 1991

### Loaders

The Funnelweb system includes special auxiliary loader files LL, SL, and User List files UL and ML, which greatly enhance the normal Editor Assembler module functions.

#### Low Memory Loader LL

Some well known utilities for the TI99/4A, 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 MiniMemory 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 Editor Assembler 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 Editor Assembler REFs are recognized, but Editor Assembler 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, the normal Editor Assembler (>2002) or Extended BASIC XML (>2000) target locations are no longer available, so LL sets PAD (>8300) as the XML address for GPL returns by selecting a >FO XML in console GROM.

#### Script File Make/Loader SL

In version 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 a 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 Editor Assembler 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 version 4.40 package is the actual Script file used for complete re-assembly and program file preparation for the 80 column 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 path name (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 path name. This instructs the ScriptLoader to supply the Funnelweb boot path name 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 path name 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 8 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 8 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 Editor Assembler 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 an 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 Extended BASIC 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 full filename (with the 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 version 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 column 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 column 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 do not have six candidates, leave the remainder as spaces.

Either of the 40 column or 80 column 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 >E980 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 version 4.40, July 1991

### Utilities

Utility files of various origins are included on the Funnelweb distribution disk(s). Files LGEN/S (for generating templates for Extended BASIC programs like LOAD) and LDSR/S for use with LowLoad from version 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 customise 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 "?" <FTN-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 version 4.0 c99 compiler files C99C, C99D, C99E 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, C99D, C99E from the same disk drive. The return from c99 reloads FW from either the set path name or the Editor Assembler 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 Editor Assembler SAVE utility loads as absolute code in low memory. Funnelweb is compatible with SAVE, but does take up its own 8K 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 Editor Assembler 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 Editor Assembler compatible memory image format.

If the Loader has placed files so that SFIRST is in high memory 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 >2878 (above the Editor Assembler 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 (Option 8) files, FSAVE saves its first module from low memory from SFIRST to the top of low memory, nominally >3FFA (at UTLTAB+4) and then from high memory from >A000 to SLAST. If SFIRST and SLAST both point to the same segment the SAVE is normal. The MBSAVE entry adjusts the high memory start to >A050 above the Mailbox. Use Editor Assembler SAVE for addresses in the >6000 to >8000 cartridge space.

The MEMSAV entry point allows direct entry of hexadecimal 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 Editor Assembler SAVE utility, amongst its other little foibles, adds a further 8 bytes to this count, but the program file loader in the Editor Assembler module believes the byte count in the header. In normal usage the extra 8 bytes, falsely indicated by Editor Assembler 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 Editor Assembler 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 version 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 Editor Assembler, MiniMemory BASICS by CALL LOAD(".."), Myarc XB-II by CALL LR(".."), Editor Assembler Load and Run, or most other object file loaders such as come with Myarc or CorComp disk controllers. It may be kept in MiniMemory cartridge RAM if you follow the MiniMemory instructions for forcing it to load there. The RUN name is LDFW. It offers choice of several path names 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 Editor Assembler or MiniMemory 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 path name load is chosen CONFIG will remind you that only the filename or further path name should be entered. There is no entry corresponding to "9 <CRT ROM> 0" which looks for a cartridge ROM header at >8000 and "9" executes the first program listed there and "0" the second. This may be handy for owners of version 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 Option 3.

(6) Cartridge RAM loader CT8K/O

Object file CT8K/O is used to store Funnelweb in >8000 to >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 operating system). 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 and Run option. Funnelweb is loaded into high memory on selection as for FW.

### (7) Assembly line locator LH

LineHunter is one of our working tools now made available for Funnelweb users, although 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 8 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) Extended BASIC FORTH Loader XB4THLD

This program allows the standard TI FORTH disk to be loaded by TI Extended BASIC. It works only with the Extended BASIC and Editor Assembler modules and its primary use would be from the Extended BASIC User List.

44  
Funnelweb version 4.40 - October 1991

### Bugs and Problems

#### (1) Bugs and Modifications

The bug report is being zeroed out at the time of first issue of version 4.40 of Funnelweb, but new bugs are being created all the while. That is 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 Extended BASIC RUN from DR80 engages Extended BASIC but then locks. <Quit> is still there though. The cause is not yet obvious.

System bug ..... Anomalous error recovery behaviour may occur with SD and both DiskReviews in systems with 80-track Myarc floppy disk controllers. See FWDOC/DR41 or FWDOC/DR81 for details.

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

Jul/28/91 ... Fixes for LOAD's Extended BASIC user list program loading, and for ED80 <P>Dir function from SD. The FILENT routine in FW/LOAD now clears the last character in the window with <FCTN-3>. ML40 repaired.

Oct/30/91 ... Back to programming with renewed vision!! Read/write failure indicators added for DiskReview file copy. DISKU notes handling revised and corrected. FWDOC files DR40, DR41, DR80, DR82, PSRV updated. Dual 80/40 editor, files ED80 and ED81 now windows correctly in 40 column mode. Cause of error recovery bug in DR narrowed to Myarc 80-track floppy disk controller.

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) Programmer's 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 TI99/4A (and level of Fairware support) it is unlikely to be reorganized. Details which may be relied on by external programmers are given in the supplementary 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 Extended BASIC user list data and 2 pieces of machine code tucked between the Extended BASIC 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 Extended BASIC program code. These are shifted to follow the normal Extended BASIC Utilities (the memory pointer is updated) and are available for use by Extended BASIC programs loaded from the Extended BASIC 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 Extended BASIC 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 (>E008) 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 Editor Assembler 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 version 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 Extended BASIC return is at >2000 and the Editor Assembler return is at >2002. All others use PAD (>8300).

With these preparations it was possible to write a GPLLNK to fit invisibly in the Editor Assembler utilities without altering the FFALM pointer. In any event the Editor Assembler GPLLNK cannot be used because it branches to Editor Assembler module GROM code. Programs run other than from Editor Assembler or Extended BASIC 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 @>70 returns.

If Funnelweb is overwritten, the simplest return is to the Title Screen. If your program does not make extensive use of VDP memory, it may be possible to stash Funnelweb (from >E980 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, (>E008) 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 TI99/4A 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 path name assumed, as appropriate. The flag to turn on boot disk tracking in FW is at BTF1AG 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 TI-Writer and Editor Assembler access is needed.

The flag for a hard path name is at HDFLAG EQU >FF02 and the path name 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 Funnelweb 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 (Options 1 to 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 Extended BASIC 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 cannot do anything about. If you have commercial programs which you would like to use with Funnelweb but cannot 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 Editor Assembler 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 disk manager program to interchange file names. File lengths of >2000 in 33 sector files, such as produced incorrectly by Editor Assembler 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 Editor Assembler 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 Editor Assembler 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 were not 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 Editor Assembler module gives the easiest way to switch back and forth between Funnelweb and a standard environment.

Some programs in Editor Assembler program format contain code to unload the standard utilities from the Editor Assembler GROM because the Editor Assembler 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 Editor Assembler. c99 is smart enough not to try getting code from non-Editor Assembler GROMs, but other programs may need revision.

Funnelweb has also adjusted the Editor Assembler 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 Editor Assembler routine. Error returns may need special attention.

#### (c) Exit difficulties

Read (4) above on returns. Option 2 loads will be necessary for program files that overwrite the Editor Assembler utility area in low memory but still use a GPL return via the contents of R11 as would be acceptable for an Option 2 (or Editor Assembler option 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 0480 0070 to 0420 0000 to return to the title screen. Use hexadecimal 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.