

RADIO SHACK EDITOR/ASSEMBLER EXATRON STRINGY FLOPPY PATCH

1.0 INTRODUCTION

The purpose of this patch to Radio Shack's Editor Assembler (EDTASM) is to allow the use of the Exatron Stringy Floppy (ESF) for the storage of source and object code. The patch (EAPAT) includes a lower-case capability (for those with the lower-case modification), keyboard debounce, and input and output to cassette tape or ESF wafer. This patch requires a TRS-80 model I level II with a 16K (minimum) system and a Exatron Stringy Floppy with ROM 4.1. An original distribution version of EDTASM from Radio Shack is also required. Version 1.1, 1.2, and Series 1 version 1.0 EDTASM distributions are supported.

2.0 INSTALLATION

The patch allows object code to be saved on an ESF wafer. Since only cassette object files are supported by the ROMs, an object code loader program must be used to load the generated ESF object code file. The installation of this patch will be made using this program (LOADER) to read the ESF patch from wafer and install it. The following is a list of the files on the distribution wafer:

1. LOADER - (BASIC command line version)
2. LOADER - (Relocatable version)
3. EAPAT 1.1 - (EDTASM 1.1 patch)
4. EAPAT 1.2 - (EDTASM 1.2 patch)
5. EAPAT S1.0 - (Series 1 EDTASM 1.0 patch)
6. TRSTST - (Example program)

Follow the instructions given below to install the patch to your version of EDTASM.

1. Insert the patch distribution wafer and type:
@LOAD 1 <ENTER> (Load the object code loader)
2. Load your original unpatched distribution version of EDTASM from either cassette or ESF wafer but do not execute it.

For cassette type:

SYSTEM <ENTER>
EDTASM <ENTER>
<BREAK>

For ESF wafer type:

@LOAD X <ENTER> - (X is your EDTASM file number)

Note: Hold down the shift key to prevent execution.

3. Determine which version of EDTASM you have (either 1.1, 1.2, or S1.0). Reinsert the patch distribution wafer and enter the following:

For 1.1 - LOAD 3 <ENTER>

For 1.2 - LOAD 4 <ENTER>

For S1.0 - LOAD 5 <ENTER>

Note: Be sure not to type an @ sign in front of LOAD.

4. Place a write enabled scratch wafer in the ESF and type the following:
 - For 1.1 - @SAVE 1,17152,7691,24325 <ENTER>
 - For 1.2 - @SAVE 1,17152,7680,24314 <ENTER>
 - For S1.0 - @SAVE 1,17990,8483,25957 <ENTER>
5. You have now patched your EDTASM. Try your new version by typing @LOAD 1 <ENTER>. On initial execution the program will display a brief description of the enhancements provided by the patch.

3.0 USING THE NEW EDTASM

The patch features keyboard debounce, lower-case, and source and object code transfers using cassette tape or ESF wafer. The LOADER programs are used to load ESF object code files. These enhancements are described separately below.

3.1 LOWERCASE

The assembler initializes with lower-case entry disabled, but lower-case display is always enabled. Lower-case entry may be enabled or disabled by pressing the CLEAR key. The cursor indicates this by changing from an underscore (upper-case only entry mode) to a graphic block (lower-case entry with upper-case using the shift key). The assembler will support lower-case in comments and assembled ASCII strings and characters. Lower-case may not be used in labels, opcodes, or EDTASM commands. The entry mode may be changed at any time by pressing the CLEAR key.

3.2 SOURCE CODE TRANSFERS

The patch allows both cassette and ESF source input and output. The EDTASM command L and W are still used but with the following format:

```
Load from Cassette L (space ) FILE NAME
Write to Cassette W (space) FILE NAME
Load from ESF L#AD
Write to ESF W#XD
```

The space after L or W designates a cassette operation, which performs as before. The # is the ESF decimal file number from 1 to 99, A is to append a file on a Load (no affect on a write designated by X as shown, but must be present), and D is the ESF drive on which the operation is to be performed. Everything after the file number is optional with no append and the last designated drive as defaults. A verify operation is performed after the source code is written to the wafer. In cassette operation a single blank after the L or W command causes the L command to load the next source file from the cassette tape and the W command to write the edit buffer with a name of NONAME to cassette tape.

3.3 OBJECT CODE OUTPUT

Object code may be written to either cassette tape or ESF wafer. After the code has been assembled with the A command the following is displayed:

ENTER FILE #

A response of a decimal number from 1 to 99 will cause the tape after the designated file to be cleared and then the object code written. Pressing the ENTER key without specifying a file number will cause the "READY CASSETTE" prompt to be displayed. The code will then be written to cassette tape.

Note: All ESF functions may be interrupted by pressing the BREAK key. Error checking and reporting is performed with the "BASIC type" error messages.

3.4 OBJECT CODE LOADER

Two ESF object code loaders are provided. File 1 is the BASIC command line version. In this version the loader is automatically relocated to the end of memory and linked to the disk basic "LOAD" command. After the LOADER has executed, you may load ESF object code files by typing:

LOAD#,D

Where # is the decimal file number and D is the drive number. The loader prints the file name, starting memory location, length in bytes, and entry point. Programs loaded may be $\text{\textcircled{S}}$ SAVED using the numbers displayed.

File 2 is a version which may be relocated to a desired section of memory. The following is displayed when initially loaded:

ENTER HEX BASE ADDRESS XXXX

Where XXXX is the four digit hexadecimal start of the target memory for the loader and also its re-entry point. After the program has been relocated the following is displayed:

ENTER FILE # #XD

Where #XD has the same format as the EDTASM W command. The file name and numbers are printed and the program is repeated. Pressing the BREAK key causes a return to BASIC. Pressing the CLEAR key causes a jump to the last entry address read. This may be used to execute a program after the program has been loaded. The loader may be re-entered at the hex relocation address.

3.5 COPYING THE LOADERS

You may copy the two loader programs by holding the shift key down while loading. @SAVE them using the numbers printed when the @LOAD finished.

4.0 EXAMPLE

File 5 on the distribution wafer is an example program which performs a simple system checkout. It should prove helpful to use this program to acquaint you with the use of your patched EDTASM. The program displays the full character set producing lower-case and graphic characters. The BASIC ROMs are then checksummed and compared with the values set by the equates at the beginning of the program. The length of the memory is then found and some tests are performed on it. The results are displayed for a while and then the program is repeated.

Load and execute the patched EDTASM and place the distribution wafer back in the drive. EDTASM will initialize with a brief description of the patch displayed as reminder of SYNTAX and as identification of the patch. Load the example program by typing:

L6 <ENTER>

After the file has loaded, place a scratch 20 foot or longer wafer in the drive. At this point you may want to edit the program to change the checksums to those for your BASIC ROMs. The source may now be written to the wafer by typing:

W1 <ENTER>

Assemble the code by typing:

A TRSTST

Write the object to the wafer by typing:

2 <ENTER>

In response to the "ENTER FILE#" prompt.

To execute the program, boot BASIC and @LOAD the manually relocatable LOADER (file 2 on the distribution wafer). Enter 7000 in response to the hex base address prompt. Place the EDTASM generated source and object wafer back in the drive. Enter 2 in response to the "ENTER FILE#" prompt. The file name, starting address, length, and entry address will be displayed. Execute the program by pressing the clear key. The program may be stopped by pressing the reset button.

To save the example program to wafer, @LOAD the BASIC command line loader (file 1 on the distribution wafer). Place the object wafer back in the drive and type LOAD 2. After the program has loaded, type @SAVE 3 followed by the three decimal numbers displayed by the loader. Now @LOAD 3 to execute the saved memory image of the program.

Note: Either loader could have been used but both were used to acquaint the user with their operation.

5.0 WARRANTY

The EAPAT wafer is guaranteed to be free from defects. If the program cannot be loaded, return it to Exatron for replacement. The EAPAT software is distributed on an "AS IS" basis without warranty. Neither Exatron nor the ESF Owner's Association shall have any liability or responsibility to customer or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by programs sold by Exatron or the ESF Owner's Association. This exclusion of responsibility and liability includes but is not limited to any interruption of service, loss of business, or anticipatory profits or consequential damages resulting from the use or operation of such computer programs.

6.0 ESF OWNER'S ASSOCIATION

The ESF Owner's Association (ESFOA) exists not only to keep you the ESF user, informed of new applications and technical information but also serves as a distributor of some of the application programs involving ESF. The goal is to provide useful and reliable software at a reasonable price. Fees charged for programs are divided between the program author and ESFOA. The funds received by ESFOA are used to finance literature, mailing, reproduction, distribution costs, and the ESF HOT LINE. By encouraging your friends to purchase their programs from ESFOA you are benefiting not only yourself as a member of the ESFOA, but you are continuing to provide the incentive to authors for the generation of more software of high quality.

Questions or comments regarding this program should be addressed to:

David Perdue
Greenview Village Apt. A440
Great Mills, MD 20634