FTL Modula-2 Editor ToolKit

CP/M Version

This disc contains the entire sources for the FTL Modula-2 editor and a number of utilities which you may find useful.

Amstrad owners should note that Side 1 of their Editor Toolkit disc contains the main editor files while Side 2 contains the utilities and the custom editor files.

Note for Upgraders

If you have upgraded from a version before 1.24 you must note that the editor now will remember the first ten error possitions that its given, This is refiected in the data passed by the compiler to the editor. Editor hacks, you must take this into account!

See README.NOW on the FTL CP/M disks for more on how this new editor feature works.

Compiling the Editor

The file RECMPED.BAT can be used to recompile the editor. You can use the following command to do this:

```
SUBMIT RECMPED [RETURN]
```

The distributed form of this file assumes that the files are all on the current drive. You may want to add `B:` (say) in front the filenames being compiled.

Note that SUBMIT only works from drive A on many computers.

Mapping your keyboard

If you want to modify the editor to use your keyboard arrow keys and function keys, you will want to edit the module KEYBOARD.MOD, then recompile the editor (see above).

Editor Cook's Guide

The main modules in the editor are:

ME The top level module.

EDITCONT Controls the screen display

EDITDISP Some other screen display procedures, essentially part of EDITCONT.

MAKEEDIT Performs the changes to the file(s) being edited.

KEYBOARD Accepts input from the keyboard and calls appropriate routines to perform the requested edit functions. This module really drives the whole editor, once it has been got going by ME.MOD.

RESETDIS Resets the disc system.

EDITSTAT Picks up the file position of a file when it's opened.

DOMENU Supports the options menu (that is, open file, comp/exit, etc). Lots of interesting modules which you may want to press into service in your own programs hang off this module, so reading this module and the modules it calls is a good way of getting into writing applications that implement a similar range of facilities.

MACROS Handles macro expansion and definition, including the learn mode. Makes and saves them *on the fly* during edits, too.

SETUPCAL Editor uses this module to call the compiler. See the module CHAIN for another example of this.

SCREENIO Does screen manipulations. See description below and the modules for more detail.

SCREENIO Details

The screen input-output routines used by the editor. The standard(terminal style) version of ScreenIO (SCREENIO.MOD) plus some memory mapped versions are supplied.

SCREENIO.OSB is for the Osborne 1, SCREENIO.MBE is for the MicroBee and SCREENIO.EAR is for the Earth Computers' Z80 card. To produce a memory-mapped version, start with the MicroBee since that is the most standard.

See the source code formore details. To replace the standard version of ScreenIO by a memory-mapped one, compile the memory mapped implementation module in place of the standard implementation module, no changes are needed in the definition module.

Note: the Osborne version is for Osborne 1s only, not the Executive. The Vixen is also slightly different.

Utility Programs Included

There are a number of useful utilities included on the Editor Toolkit disc; some of them are also supplied on the main compiler disc and documented in the main manual, others are new to the Editor Toolkit. These new files are:

CAT

CAT is a simple listing program. It was intended originally for use on a bulletin board system in place of TYPE. It can, however, do some things which TYPE cannot:

Number output lines,
Page output,
Expand tabs to spaces,
Wrap lines for listing to terminals without auto line-wrap,
Reposition by line number or by string search,
Use wild cards on the command line,
See non-printing characters in hex (other than CR/LF/TAB)

It is not a CP/M implementation of the cat program on UNIX; it is more like UNIX's more.

The calling sequence is:

CAT file list /options

or

CAT file list [options

The options may be omitted. file list is a list of filenames or wild cards which match the filename.

For example:

```
CAT b:*.def b:*.mod
```

will list the definition files, then the module items.

The options are:

I Expand tabs. Tabs are expanded to an appropriate number of spaces.

N Number lines. Each line is preceded by a line number in the listing if this option is used.

P Use page mode, can be followed by lines per page.

W Set software line-wrap. Lines longer than the line width will be wrapped around by the software. This is useful if your terminal will not perform hardware line-wraps. It also prevents any text falling off the top of the screen as a result of hardware line-wraps. W may be followed by the number of columns on the screen. Default width is 80.

Examples:

```
CAT b:*.def/NP
```

Use page mode and number the lines.

```
CAT b:*.def/P16W64N
```

Use page mode and software wrap with pagelength 16 and width 64.

When in paged mode, you can enter commands at the end of each page. In fact, you can enter a command character at any time.

At the end of a page of output, CAT will stop and ask for a command. You may then enter any of the commands described below. In fact, you can enter a command character at any time.

When CAT determines that a character has been typed, it will bring up the `Command :` prompt, display the character you typed, and go into page mode. Multiple commands may be placed on a line; the commands available are given on the next page:

? Display some help information.

number

Restart listing from given line number.

+ or - number

Advance (or retire) by the given number of lines.

/string/

Restart listing from the next occurrence of string. The delimiters (//) can be any printing, non-alphanumeric character. A number and a search string can be combined to search from a given line number. If no string is entered ("//"), the previously-entered `string` is searched for again.

N Toggle line numbering option.

I Toggle tab-expansion option.

W Toggle software line wrap option. If the W is followed by a number, the number resets the screen width.

P Toggle page mode. If P is followed by a number, it resets the page length. Note that the first P on a command line always causes page mode to be cleared, as page mode is always enabled when the command line is processed. Hence, to just change the page length, you must do `P20P` (say).

Q Quit this file.

[CTRL]-C
Return to CP/M.

COMPARE

An Ascii file comparison program which does line-by-line comparisons of files. It uses a powerful comparison method which rarely gets lost and which can detect block moves.

This program displays the differences between two files. It presents the output in a reasonably readable form (it would look beaut on a colour printer). To run the program:

```
COMPARE old file,new file[,List file]
```

The third file is optional. If omitted, the output is to the printer. You can use device names (e.g., TTY for the console [screen]) as well as disc file names. Lines which have been deleted from the old file are printed in italics. Inserted lines are printed in bold. Lines which have been moved to a new position are printed in their new position surrounded by lines of asterisks. You may need to patch the printer control codes for your printer. These are very near the beginning of the program (see the .MOD file). Each control code can be up to 8 characters long and is terminated by a value of 0 if less characters are required. Naturally, you may also change the .COM file.

The control characters in the supplied program are set for a **STAR DPS510**, which is supposed to be Epson compatible, so the program may well run *as is* on Epsons and Epson clones.

Operation of COMPARE

For each line, a hash total is calculated. Then the unique lines in each file are identified. When a unique line matches a unique line in the second file, the two lines are cross-linked.

After all unique lines have been cross-linked, the program 'grows' the links outwards by cross-linking pairs of lines that match and are next to a pair of cross-linked lines. Finally the text is scanned and printed out. Lines in the new file which are not linked to anything in the old file are insertions. Unlinked lines in the old file are deletions. Lines in the new file which link to out-of-order old file lines are parts of a block move.

A number of improvements could be made to this program: often, a new version of a program is created (say, for another machine) and then the original version is modified. The problem is then inserting the new changes to the original file into the other version, while retaining any changes made to produce that version. Probably fairly difficult to do. Sometimes, one of the files is on another machine. It would be nice to have the old file on the remote machine and only transfer the checksums and the lines which have been deleted across the link. Fairly easy, but reliability of the link will cause problems.

See also COMPARE.MOD itself.

COMPDIR

CompDir compares two disc directories. It is invaluable to software developers who need to ensure that their distribution discs are complete and contain the latest versions of files.

The comparison may use the full file name, just the base name, or (not very useful) just the extension. Optionally, you may create a file containing a list of the file names added, deleted or matching. The command to call the program takes the form:

```
COMPDIR wild card, wild card [, list file] [/options]
```

Where each wild card produces one of the lists to compare. To compare two discs:

```
COMPDIR a:*.* b:*.*
```

Options are any of the following (the options may be combined):

T Compare only file extension (by default the whole name is compared).

N compare only file name-base name, minus the extension. T and N are mutually exclusive.

C If file names match, compare the file contents as well.

E Output file names which match to list file.

A Output file names which have been added to the second list.

PATTERN

This is a pattern matching module which matches complex (even recursive!) patterns. See PATTERN.DEF for description of use, and TestPat.MOD as an example.

IMPORT

Lets you write utilities that read .SYM files. The assembler uses it.

We hope you enjoy using the FTL Editor Toolkit. If you have any problems, please feel free to write to use explaining your difficulty carefully with sample programs if possible. Alternatively you can phone for technical support between 3pm and 4pm Monday to Friday.

HiSoft

The Old School, Greenfield, Bedford, MK45 5DE

Tel: (0525) 718181