Increased TEX Efficiency Using Advanced EDT Editor Features

Linda Williams & Linda Hall
The University of Tennessee Space Institute, Tullahoma, Tennessee 37388
615-455-0631. Bitnet: williams@utsiv1 Bitnet: hall@utsiv1

Abstract

TEX is implemented on various computer systems often with little or no consideration given to the time saving quality of an efficient editor. For both the novice and expert TEX user, employing advanced EDT editor features greatly enhances the efficient use of TEX. Because there are volumes describing these various tools, it becomes necessary to reduce this vast amount of material to a manageable subset. This paper describes selected EDT features, such as editor initialization files and other commands which, when mastered by the user, enhance the editing power of TEX.

Introduction

This paper describes a text editor, EDT, supported by a VAX 11/785, running the VMS operating system. Under the name EDT+, this versatile editor will also operate under MS-DOS (2.0 or higher) or UNIX on Ardent, AT&T, Celerity, Convergent, DEC, Encore, Gould, Hewlett-Packard, IBM, Intergraph, MIPS, SUN, and many others.

Because skill levels in using EDT as well as other editors vary greatly, from novice to expert, advanced editor features are often overlooked. Although there are volumes of documentation on editors, with help files and computer-aided instruction, learning about these advanced features often occurs by passing information from user to user. It is the purpose of this paper to continue that tradition by describing three fundamental areas of EDT, namely, 1) EDT line-mode features, 2) EDT screen-mode keypad features, and 3) EDT initialization features. Other editors assign similar type names to their editor features; keyboard macros, learning keystrokes, or recording keystrokes. Whatever your editor, the message is still the same: when an efficient editor is mastered, knowledge of its features and capabilities give the user a fast and powerful method of editing.

EDT Line-mode Features

When working within a document, EDT offers the user two fundamental editing modes, line-mode and screen-mode. (Screen-mode will be explained in the next section.) When the edit session begins, EDT starts by presenting *, the line-mode prompt. At this prompt it is possible to use many line-mode commands or to execute a specific line-mode command called CHANGE:

*C ⟨*carriage return*⟩

This particular command allows for full screen editing capability (screen-mode). Other commonly used line-mode editor functions for creating TEX and other data files are:

1) carriage return — present the current line,
2) TYPE [range] — display the specified lines on the terminal,
3) SUBSTITUTE — replace the next occurrence of an old string with a new string over a range of specified lines,
4) WRITE [range] — write a buffer or a segment of a buffer to a different disk file,
5) DELETE [range] — delete a line or range of lines,
6) EXIT — end the editor session and save a copy to the MAIN buffer,
7) QUIT — end the editor session without saving any changes,
8) CHANGE [line #] — change to screen mode at an optional location.

Using the key sequence Ctrl Z will evoke the line-mode option * anywhere within the text for use of the above commands. GOLD (PF1) and 7 (keypad), in screen-mode will also prompt for line-mode options. Specific words and numbers can be used as qualifiers in conjunction with SUBSTITUTE and DELETE. These options allow the user one more

level of choice to prevent possible damage to the file.

The command

```
*sub/Tex/\TeX\/wh/query
```

causes the string of letters Tex to be replaced by the string \TeX\ throughout the whole (wh) file and prompts with a question mark to verify the operation on each line displayed. Possible responses to this option are yes (y), no (n), or quit (q).

The command

```
*del 121:131
```

deletes lines 121 through 131 of the file.

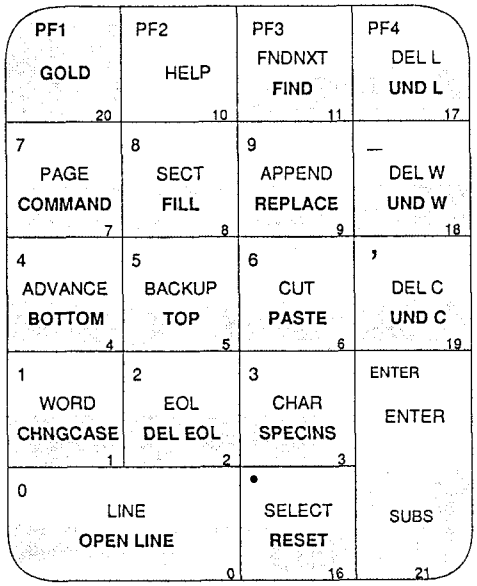
The CHANGE command makes rapid correction of errors much easier since TEX error messages are given by line number. For example, if an error was detected at line 102 in the TEX data file, it is possible to use the command

```
*c 102
```

to place the user at the point of the TEX error.

Line-mode commands provide sufficient editing power to create most TEX documents; however, by employing both line-mode and screen-mode features time-consuming procedures can be reduced. The next section will explain how screen-mode commands reduce the input for a given operation from a line-mode command word, possibly from 4 to 7 keystrokes to a keystroke sequence of one or two keys. This keystroke reduction enhances editing power.

EDT Screen-mode Keypad Features



The figure illustrates the EDT screen-mode keypad layout with associated keypad commands for VT100-type and PC enhanced-type keyboards. These EDT features are also available on other keyboards; specific key assignment sequences may vary.

Most keypad keys have two editing functions associated with them, primary function commands and alternate function commands. The primary function commands, not highlighted, use the indicated key only. The alternate function commands, highlighted, use the GOLD key (PF1) in conjunction with the indicated key to perform the desired commands.

These keypad functions can be broken down into three groups: 1) movement, 2) delete and recover, and 3) key definitions and redefinitions. The following three subsections discuss this in more detail.

Movement. Movement refers to three possibilities, 1) the general movement of the cursor, 2) repositioning the visual display to another segment of the file, for example, the beginning or the end of the buffer, and 3) the movement of the cursor to the location of a specific text string.

For general cursor movements, the 1 (WORD) on the keypad moves the cursor from the beginning of one word to the beginning of another. The 2 (EOL) on the keypad moves the cursor from the current cursor position to the end of a line.

For repositioning the screen display, the sequence of GOLD (PF1) and 4 (keypad) advances to the bottom of the text and GOLD (PF1) and 5 (keypad) advances to the top of the text. For moving the cursor to a specific text string, the command FIND uses GOLD and PF3. The command prompts the user with Search for: at which point the user types the desired string and hits 4 (BOTTOM) or 5 (TOP) to indicate the direction of the search. The cursor will then be moved to the first occurrence of the specified string in the indicated direction. Using the PF3 (FNDNXT) command will search for the next occurrence of the previously specified string.

Delete and recover. Delete and recover refers to two possibilities, 1) the complete removal of text, and 2) the movement of sections of text from one location to another. Deletion can be thought of in terms of the entities that the editor understands, the character, the word, the line, and a section of text specified by the SELECT command. In each case, the system associates a buffer with each of these types to hold the given removed entity — character, word, line, or section. Consequently, there is a delete

character buffer, a delete word buffer, a delete line buffer, and section buffer (specifically called the PASTE buffer). Since each associated buffer contains the last piece of text removed, this fact permits the recovery of that same text removed by that last command.

For the complete removal of text, there are three commands: 1) DEL C (keypad ,) — deletes the current character under the cursor, 2) DEL W (keypad -) — deletes from the current cursor position to the beginning of the next word, 3) DEL L (keypad PF4) — deletes text from the current cursor position to the end of the line. The main purpose in using these keypad commands to delete, instead of the delete key, is the previously stated advantages of the associated buffers. Using the GOLD key (PF1) in conjunction with the specific delete key undoes the specific operation. For example, GOLD (PF1) and PF4 recover the last deleted line.

For the movement of sections of text from one location to another, the CUT-and-PASTE operation is used. In conjunction with the SELECT operation, a section of text can be marked and either deleted permanently or moved to another portion of the buffer to be PASTED into the desired location. This is often useful in using similar TEX constructs. Since CUT and PASTE can also be done between multiple buffers, this offers additional power in copying previously successful sequences of TEX commands which have produced output in the desired form.

Key definitions and redefinitions. An extension of the power of EDT's screen-mode keypad features, key definitions and redefinitions enable the user to create, redefine, or relocate the operation of a previously defined or previously undefined key. Using two line-mode commands, SET and DEFINE KEY, the user is able to design complex, tailor-made keystroke sequences to perform the desired operation. The next section will illustrate the use of initialization files containing these complex sequences which can broaden keypad editing to encompass the entire keyboard.

EDT Initialization Files

When a user starts an editing session, EDT searches for a system-wide initialization file. If no system-wide command file is found, EDT then looks for a file (edtini.edt) in the default directory. Because initialization files are not required, this feature's potential power is often overlooked by the less experienced user.

The edtini.edt file is allowed to contain only line mode commands, previously stated above. The most common commands are SET and DEFINE KEY. These two basic commands allows users to customize their editing environments. The keyboard can be rearranged for specialized editing functions for individual preference reducing errors and allowing for more advanced application development.

A novice's initialization file may simply contain one line — SET WRAP 75 — forcing words hitting the 75 column to wrap to the next line. However, as expertise with the editor grows, so does the size and functionality of the initialization files. The following presents a fully commented initialization file which can be used in the creation of TEX documents; these specific commands can be implemented independently or in conjunction with other initialization files:

```
! Start-up commands for editor EDIT/EDT
      (VAX/VMS)
!
! Key definitions for EDT commands:
! set wrapping to 75 columns
SET WRAP 75
! set full-screen editing mode (KEYPAD MODE)
SET MODE CHANGE
! make searches distinguish between uppercase
      and lowercase characters
SET SEARCH EXACT
! to use with slow modem at home
DEFINE KEY CONTROL P AS "EXT SET LINES 6; SET
      CURSOR 1:4."
DEFINE KEY CONTROL @ AS "EXT SET LINES 22;
      SET CURSOR 7:14."
! B -- back to main buffer
DEFINE KEY CONTROL B AS "EXT EXTERN."
! F -- find a file to be put in buffer
DEFINE KEY CONTROL F AS "EXT INCLUDE ?'Enter
      file name: ' BUFFER EXTERNAL."
! includes header file into main buffer
      automatically
DEFINE KEY CONTROL U AS "EXT INCLUDE
      header.tex  BUFFER MAIN."
! includes file typed at the prompt
      automatically into main buffer
DEFINE KEY CONTROL I AS "EXT INCLUDE ?*
      'Insert file:'."
! R -- returns to file in buffer
DEFINE KEY CONTROL R AS "EXT RETURN."
! set screen to 132 columns
DEFINE KEY CONTROL W AS "EXT SET SCR 132."
! set screen to 80 columns
DEFINE KEY CONTROL E AS "EXT SET SCR 80."
! change case of first letter of current
      word (left of cursor)
DEFINE KEY 8 AS "-W +CHGCC -C +W."
! move cursor to next section (16 lines)
DEFINE KEY 3 AS "16L."
DEFINE MACRO EXTERN
  FIND=EXTERN
  I;FIND=EXTERNAL.
```

```
DEFINE MACRO RETURN
  FIND=RETURN
  I;FIND=MAIN.
FIND=MAIN.
```

In the creation of initialization files, or when using features from other initialization files, some commands must be used with caution; others will require further careful study to understand and utilize their full potential. For example, this illustration contains no defined commands either to exit an edit session CTRL/Z *exit or to quit an edit session CTRL/Z *quit. The possibility of mistakenly QUITing an edit session, thus not saving edit changes, rather than EXITing and saving these specific changes, is too great to risk using a previously defined key which might be hit by mistake. Other users have kept these definitions in their initialization files and have mistakenly destroyed the results of an entire edit session, a very costly result. Initializations files may also contain TEX commands, such as

```
DEFINE KEY GOLD B AS "I{\bf }^Z -C."
DEFINE KEY GOLD C AS "I\centerline{}^Z -C."
DEFINE KEY GOLD D AS "I\baselineskip ."
DEFINE KEY GOLD F AS "I\footnote{}{^Z -2C."
DEFINE KEY GOLD G AS "I{\it }^Z -C."
DEFINE KEY GOLD H AS "I\halign{}^Z -C."
DEFINE KEY GOLD I AS "I\item{}^Z -C."
DEFINE KEY GOLD L AS "I\line{}^Z -C."
DEFINE KEY GOLD M AS "I\magnification=."
DEFINE KEY GOLD N AS "I\hfill."
DEFINE KEY GOLD P AS "I\par \noindent."
DEFINE KEY GOLD Q AS "I\hskip ."
DEFINE KEY GOLD S AS "I\ss ."
DEFINE KEY GOLD V AS "I$$\vbox{}$$^Z -3C."
DEFINE KEY GOLD Z AS "I\bye."
DEFINE KEY GOLD _ AS "I\underline{}^Z -C."
DEFINE KEY GOLD , AS "I\matrix{\cr}^Z -4C."
DEFINE KEY GOLD ; AS "I{\hbox{}}^Z -2C."
DEFINE KEY GOLD : AS "I\pmatrix{\cr}^Z -4C."
DEFINE KEY GOLD ^ AS "I\eqalign{\cr}^Z -4C."
DEFINE KEY GOLD ~ AS "I\eqalignno{\cr}^Z -4C."
DEFINE KEY GOLD = AS "I\widehat ."
DEFINE KEY GOLD + AS "I\widetilde ."
DEFINE KEY GOLD \ AS "I\pmb{$$}^Z -2C."
DEFINE KEY GOLD [ AS "I\overline{}^Z -C."
DEFINE KEY GOLD ] AS "I\cases{\cr}^Z -4C."
```

The DEFINE KEY command allows commonly used TEX commands to be inserted into a TEX data file using only the defined keystrokes, e.g., GOLD C instead of typing \centerline, etc. The command is placed after the "I (insert), ^Z signals the end of the command, the curser can be moved back by specifying spacing -4C, and ." indicating placement where the command is invoked. By defining commands in this manner TEX formatting errors and typing errors can be reduced. A well-designed EDT initialization file can enhance EDT's power further in the creation of TEX.

Conclusion

This paper presents of three major topics: 1) EDT line-mode features, 2) EDT screen-mode keypad features, 3) EDT initialization files. As the paper progresses from topic to topic, each step has in effect reduced the number of keystrokes and increased the power of the editing tool. While it is sufficient to know only one editing modality, such as line-mode, familiarity with screen-mode will enhance editing even more. By adding the power of the third topic, time-saving advanced applications can be developed.

It is not within the scope of this paper to condense volumes of user manuals to just a few pages, but it is sufficient to say that there is no substitute for learning through real-life experience. Thus, the purpose of this paper is to acquaint the user with some often overlooked editing features. Efficient use of a sophisticated editor makes a fast and effective tool and its power should not be overlooked.

Acknowledgements

Without the help and patience of two individuals this paper would have been an impossible task. Our thanks goes to Harry Ferber II for his writing and presentation experience and Heinrich Senge for his ingenious .edt files.

Bibliography

Boston Business Computing, Ltd. *EDT+*. Andover, Massachusetts, 1990.

Digital Equipment Corporation. *Guide to VMS Text Processing*. Maynard, Massachusetts, April 1988.

Digital Equipment Corporation. *VAX EDT Quick Reference Guide*. Maynard, Massachusetts, September 1984.

Digital Equipment Corporation. *VAX EDT Reference Manual*. Maynard, Massachusetts, April 1988.

The University of Tennessee Computing Center. *VAXCluster User's Guide*. Second Edition, UT Publication Number E01-9915-004-89, Knoxville, Tennessee, 1989.