

CC-245-1, MAC-M-195-1

COMPUTATION CENTER
Massachusetts Institute of Technology
March 15, 1965

TO: All CTSS Users
FROM: R. C. Daley and C. Garman
SUBJECT: ED, A Context Editor for Card Image Files

INTRODUCTION:

ED is a command for editing 14-word BCD card image files within CTSS. The command is based on TYPSET (CC-244, MAC-M-193 by J. H. Saltzer) and many of the conventions of TYPSET are used by ED. Tabs are automatically interpreted for FAP, MAD, MADTRN, LISP, COMIT, and ALGOL (i.e., AED) programs. Tabs may also be set by the user for other purposes. Although line numbers may be generated by the ED command, editing is done entirely by context. The ED command is offered as an alternate to the present INPUT, EDIT and FILE commands.

USAGE:

The ED command is initiated with the following CTSS command.

```
ED NAME1 NAME2 NAME3
```

NAME2 is the secondary name of the file to be edited or created and must be provided if ED is to set the tabs. NAME1 is the primary name of the file to be edited. If NAME1 NAME2 is not specified, ED will assume that a new file is to be created and will start in the high-speed INPUT mode. If NAME1 is provided, the command will look for the file NAME1 NAME2. If the file is not found, the high-speed INPUT mode will be entered. If the file is found, the EDIT mode will be entered.

If NAME3 is specified and the file NAME1 NAME2 is found, the subsequent FILE will create a file NAME3 NAME2. NAME1 NAME2 will remain unaltered. Any arguments to the FILE request, however, will take precedence.

HIGH-SPEED INPUT MODE:

When the user enters this mode, the ED command will type "INPUT:" on the user's console. While the user is operating in this mode, the ED command will accept input lines from the user's console. Tabs will be interpreted automatically for each input line. Backspace characters may also be used to move back one character position in the input line. No response is typed for input lines and as a result, the user may type successive lines as fast as he wishes. When the user types a line consisting only of a single carriage return, the ED command will place the user's console in the EDIT mode.

EDIT MODE:

When the user enters this mode the response "EDIT:" will be typed on the user's console. At this time the user may type requests to the ED command. All changes made to a file become effective immediately and as a result, the user is able to make recursive modifications to his file. Conceptually we may think of a pointer which is positioned at a line in the edited file. When the user enters the EDIT mode from the INPUT mode, this pointer will be positioned at the last input line typed by the user. When the user starts the ED command in the EDIT mode, the pointer is positioned before the first line in the old file. If the end of file is reached by an EDIT request, the comment "END OF FILE REACHED BY" is typed on the user's console followed by the request which caused the end of file to be reached. At this time the pointer will be positioned after the last line in the file. When in the EDIT mode, any line which is not a legitimate EDIT request will cause the comment "NOT A REQUEST" to be typed on the user's console followed by the line which caused the error. In many cases it is possible for the user to stack EDIT requests. If one of the requests causes an error message to be typed, any stacked requests will be ignored. This is done in case one of the stacked requests depended on the successful completion of the request in error.

Any number of initial tabs or spaces (including 0) may occur in a request line. Arguments and the request may be separated by at least one space or any number of tabs or spaces. Wherever, the argument is text, however, tabs and spaces retain their normal significance.

Warning:

One interrupt is set which returns the user to EDIT mode. Use it sparingly, as it can result in lost lines in files if used during any request which moves the pointer (FIND, LOCATE etc.).

EDIT REQUESTS:

```

REQUEST:      FIND LINE
ABBREVIATION: F
RESPONSE:     none
ERRORS:       END OF FILE

```

The FIND request is used to move the pointer forward from its present position to the line specified by LINE. LINE is a normal input line and may contain tabs and backspaces. This line is used as a mask for selecting the desired line in the edited file. Matching is done only on the non-blank characters specified in LINE. For example, the request,

```
F (tab)~(tab)-ALPHA,1
```

might be used to find the line,

```
LOOP    TIX    ALPHA,1,4
```

```

REQUEST:      LOCATE string
ABBREVIATION: L
RESPONSE:     none
ERRORS:       END OF FILE

```

The LOCATE request is used to move the pointer forward from its present position to the first line which contains the entire character string specified by "string". The full line of 84 characters is scanned, so that "string" may specify line numbers. It is recommended that "string" include the leading zeros of the line numbers to avoid any undesired match with program constants.

```

REQUEST:      NEXT I
ABBREVIATION: N
RESPONSE:     none
ERRORS:       END OF FILE

```

This request is used to move the pointer forward from its present position in the file. "I" specifies the number of lines to be skipped over. If I is "0" or not specified, it is assumed to be "1" and the pointer will be moved to the next line in the file. If the NEXT request is given after the end of file has been reached, the pointer will be reset to the beginning of the file and moved "I" lines from there.

```

REQUEST:      DELETE I
ABBREVIATION: D
RESPONSE:     none
ERRORS:       END OF FILE

```

The DELETE request will delete "1" lines from the file starting with the line at which the pointer is currently positioned. The pointer is left at the position vacated by the last line deleted by this request. If 1 is "0" or left unspecified, only the current line will be deleted.

REQUEST: PRINT 1 L
 ABBREVIATION: P
 RESPONSE: printed lines
 ERRORS: END OF FILE

The PRINT request will print "1" lines from the file starting with the line at which the pointer is currently positioned. Upon completion of this request, the pointer will be left pointing to the last line printed. If 1 is "0" or left unspecified, one line will be printed. Normally lines are printed without line numbers. If the character "L" is present in the PRINT request, line numbers will be printed to the right of the printed lines.

REQUEST: RETYPE LINE
 ABBREVIATION: R
 RESPONSE: none
 ERRORS: none

This request will cause the line at which the pointer is currently positioned to be replaced by LINE. LINE is a normal input line and may contain tabs and backspaces. The pointer is not moved by this request.

REQUEST: TOP
 ABBREVIATION: T
 RESPONSE: none
 ERRORS: none

This request will cause the pointer to be reset and positioned before the first line in the file.

REQUEST: BOTTOM
 ABBREVIATION: B
 RESPONSE: INPUT
 ERRORS: none

This request will cause the pointer to be positioned after the last line in the file. Upon completion of this request the user's console will be placed in the high-speed INPUT mode. All subsequent lines will be treated as input and added to the end of the file.

REQUEST: INSERT or (C,R.)
 ABBREVIATION: I
 RESPONSE: INPUT:
 ERRORS: none

This request will cause the user's console to be placed in the high-speed INPUT mode. All subsequent lines will be treated as input and inserted after the line at which the pointer is currently positioned. If the INSERT request is given immediately following a TOP request, the inserted lines will be placed at the beginning of the file.

REQUEST: INSERT line
 ABBREVIATION: I
 Response none
 Errors: none

The INSERT request may be used to insert a single line without changing to the high-speed input mode. Line is a normal input line.

REQUEST: CHANGE Qstring1Qstring2Q I G
 ABBREVIATION: C
 RESPONSE: none
 ERRORS: END OF FILE

This request will examine "I" lines starting at the line at which the pointer is currently positioned. Upon completion, the pointer will be left positioned at the last line examined by this request. If I is "0" or left unspecified, it is assumed to be "1" and only the current line will be examined. The character "Q" is taken to be the delineator or "Quote character" and may be any character in the 6-bit BCD set. "string1" and "string2" are arbitrary BCD character strings and may be of different lengths. If the character "G" (GLOBAL) is present, every occurrence of string1 will be replaced by string2. If "G" is not present, only the first occurrence of string1 will be replaced by string2 in each examined line. EXAMPLES:

line: ALPHA= ALPHA+ALPHA
 request: C 'ALPHA' BETA'
 new line: BETA= ALPHA+ALPHA
 request: C 'ALPHA' DELTA' I G
 new line: BETA= DELTA+DELTA
 request: C 'DELTA+'
 new line: BETA= DELTA

REQUEST: BLANK line
 ABBREVIATION: BL
 RESPONSE: none
 ERRORS: none

The BLANK request will put blanks in the current line wherever non-blank characters appear in "line", for example 'BL⁰' will clear the label field of a line in a FAP file.

REQUEST: OVLAY line
 ABBREVIATION: O
 RESPONSE: none
 ERRORS: none

The OVLAY request will place the non-blank characters of "line" into the corresponding position of the current line. Notice that only non-blank characters of "line" replace what was in the current line. For example in a FAP file, if the current line is

			TXI		*+1	
then.						
	0	EOF(tab)	bbH	(tab)(tab)		comment
will produce		EOF	TXH		*+1	comment

REQUEST: VERIFY
 ABBREVIATION: VE
 RESPONSE: none
 ERRORS: none

The VERIFY request sets the verify mode. In the verify mode, the requests FIND, NEXT, LOCATE, OVLAY, BLANK and CHANGE will cause the printing of the current-pointer line. In addition, CHANGE will cause the printing of all changed lines. Requests may not be stacked while in the verify mode.

REQUEST: BRIEF
 ABBREVIATION: BR
 RESPONSE: none
 ERRORS: none

The BRIEF request sets the brief or normal mode. Within the brief mode, the FIND, NEXT, LOCATE, OVLAY, BLANK, CHANGE requests will give no response.

REQUEST: CLIP 'ON' or 'OFF'
 ABBREVIATION: CL
 RESPONSE: none
 ERRORS: TRUNCATED:

The request CLIP ON sets a mode such that any input line which exceeds column 72 will cause the message "TRUNCATED:" followed by the faulty line image. Any waiting input lines will have been deleted. Requests on which this may occur are FIND, INSERT, RETYPE, OVRLAY, BLANK and high-speed INPUT. The request CLIP OFF resets the mode to its normal setting.

REQUEST: SERIAL N
 ABBREVIATION: S
 RESPONSE: none
 ERRORS: none

This request is used to change the increment between line numbers of successive lines to the increment specified by the decimal integer "N". Initially, this increment is set to 10 by the ED command. If N is "0" or not specified, it is assumed to be "10". Lines inserted after a line with the line number "L" will be sequenced L+N, L+2N, L+3N, etc. If the lines following the inserted lines have line numbers which are less than or equal to the line number of the last inserted line, as many lines as necessary will be resequenced to insure that all line numbers are unique and in ascending order. For example, assume that "N" is 2 and the user wishes to insert 9 lines after line 25 in a file that was previously sequenced by fives. The inserted lines would be numbered, 27, 29, 31 ... 43. The lines previously numbered, 30, 35, 40, 45 and 50 would be renumbered to, 45, 47, 49, 51 and 53 respectively. The remaining lines in the file would be unchanged.

REQUEST: COLON a
 ABBREVIATION: CO
 RESPONSE: none
 ERRORS: illegal argument

A colon (or backspace on 1050) is a logical backspace anywhere eg., 'ABC ::: Db(C.R.) is interpreted as 'DbC'. The colon moves the character pointer back one but does not erase the characters over which it has moved. One should be careful in using this convention that the total number of characters does not exceed 84, as any extras will be added to the next line during INPUT, or result in a request during EDIT.

The COLON request allows the colon character to be inserted as text. (They may also be 'CHANGE'd in as desired.) If 'a' is T or TEST, all ':' will be treated as text except for the ':' as

the first character after a tab. If 'a' is B or BACKUP the normal mode will be reinstated and all 'a' will be backspaces.

```
REQUEST:      TABSET T1 T2 ... TN
ABBREVIATION: TA
RESPONSE:     none
ERRORS:       ILLEGAL TAB SETTING
```

T1 specify the columns at which tabs are to be set. Tabs must be set in ascending order and may not exceed column 84.

```
REQUEST:      FILE ~ NAME1 ~ NAME2 ~
ABBREVIATION: FL
RESPONSE:     Ready message from CYSS
ERRORS:       NO FILE NAME GIVEN
```

This request is used to terminate the editing process and write the new edited file on the disk. NAME1 specifies the primary name of the new file to be created. If NAME2 is not specified, the secondary name of the original file or the ED command NAME2 argument will be used. If NAME1 NAME2 is not specified, the old file will be replaced by the edited file. If no name was given by the initial ED command or by the FILE request, an error message will be printed and the FILE request will be ignored.