The Hessling Editor
Version 3.1
Reference Manual

Table of Contents

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual Version 3.1

Table of Contents

THE Reference Manual
Version 3.1

INTRODUCTION

THE is a text editor that uses both command line commands and key bindings to operate. It is intended to be similar to the VM/CMS System Product Editor, XEDIT and to KEDIT from Mansfield Software.

THE was originally written to be used by people already familiar with the above editors. For this reason, this document provides limited information on using THE, and concentrates more on reference material, such as command syntax and configuration.

LICENSE

THE BASICS

The default screen displayed when THE starts consists of several areas:

- a window which displays the contents of the file being editted. This is the filearea ,
- a command line from which commands may be issued,
- a prefix area which shows line numbers and from which prefix commands can be issued
- an idline which displays the file name, row/col etc. for the current file and
- a status line which indicates global status info like number of files being editted, time etc.

When THE starts, the cursor is positioned on the command line . To move between the command line and the filearea , the CURSOR HOME command is used. By default this command is bound to the HOME key (under DOS and OS/2), the DO key (on vt220s) and HOME (on xterms).

To execute commands from the command line simply type the command and press the ENTER (or RETURN) key.

To determine what keys are bound to what commands, execute the SHOWKEY command from the command line . As you press each key, THE will respond with the name of the key and any commands bound to that key. To exit from the SHOWKEY command, press the spacebar.

Key bindings may be changed for the current session by using the DEFINE command. To keep key bindings between sessions, the DEFINE commands can be placed in a profile file, which is executed each time THE starts. For more information on this, see the next section; PROFILE FILE.

It is possible to make THE look and behave more like either XEDIT or KEDIT. See the SET COMPAT command for further information.

As mentioned before, this document provides little tutorial information. For those users who have a REXX enabled version of THE, a self−running demonstration macro is supplied which will provide a better explanation of the capabilities of THE, than any documentation could. To run this demonstration, start THE as follows:

```
        the −p demo.the demo.txt
```

COMMAND−LINE SWITCHES

THE recognises many command−line switches when starting THE. All switches are specified with a single − followed by a single character. The case of this character is relevant. Some switches take extra parameters; those that take optional extra parameters are indicated by the extra parameter being stated below in *[]* . The purpose of each of the switches is as follows:

- −n Run THE without any profile. Normally THE tries to find a profile file and execute this on startup. This switch suppresses that search and execute and runs THE in its default mode. This switch is useful for determining if a bug in THE occurs only with certain user−specific customisations.
- −m On ports of THE that support colour, this switch forces THE into monochrome mode. More a testing feature than a user feature.
- −r This switch enabls THE to be run in readonly mode. In this mode commands that alter the contents of a file are invalid.
- −s On Unix platforms, this switch enables the writing of a core file if THE crashes. Normally, THE traps any internal errors and exits gracefully. This switch is a testing feature rather than a user feature.
- −b When you want to use THE as a non−interactive tool for manipulating the contents of one or mode files, this switch will disable any display of file contents and disable keybord interaction. Normally used in conjunction with a specific profile; see −p option.
- −q Run quietly in batch mode. This will suppress the introductory informative message displayed when errors are encountered running in batch.
- −k[fmt] Allows *soft label keys* . This allows the display of one (or two) lines at the bottom of the screen with *buttons* intended to be used to represent function keys. These *soft label keys* can be manipulated with the SET SLK command. The *fmt* optional extra argument is a single digit representing the format of the display of the *buttons* .

  1 − displays 8 *buttons* in a 4−4 layout
  2 − displays 8 *buttons* in a 3−2−3 layout
  3 − displays 12 *buttons* in a 4−4−4 layout
  4 − displays 12 *buttons* in a 4−4−4 layout with an index line
  5 − displays 10 *buttons* in a 5−5 layout

  Not all platforms support all 5 format options. On those ports of THE that are mouse−aware, the mouse can be pressed on a *button* , and the command assigned the the coresponding function key is executed.

- −l line This switch specifies the line number to make current when THE starts.
- −c col This switch specifies the column number to make current when THE starts.
- −p profile Specifies the THE profile to run instead of the default profile. See next section for further details.
- −a profile_arg Specifies the arguments that are passed to the profile specified with the −p switch.
- −w width Specifies the maximum line width for a line in the current edit session. Can be overridded with the SET WIDTH command.
- −u display_width THE can run as a binary editor. Specifying this switch tells THE to read in the file and display it in *lines* that are *display_width* long. All end−of−line characters in the file are ignored and are treated as other characters in the file.
- −X X11_switches With the X11 port of THE, standard X11 switches can be specified with this switch to dynamically configure the way THE displays or behaves. You can also specify XCurses−specific switches here as well. For more information on the XCurses switches available, consult the PDCurses documentation.
- −1[dir] Tells THE to run in *Single Instance Mode* . The first time THE is run with the −1 switch, its starts as normal. Subsequent executions of THE with the same −1 command−line switch will not start a new instance of THE, rather it will edit the file(s) specified on the command−line in the currently running instance of THE. You can optionally supply a directory with the −1 switch to specify where the THE FIFO (.thefifo) is created. If not supplied, the THE FIFO is created in the users $HOME directory. This feature is currently only available with the X11 port of the using XCurses 2.5 and greater. If the first instance of THE with the −1 switch crashes for any reason, subsequent attempts to run THE with the same −1 switch will hang. To fix this remove the FIFO and then run THE with the −1 switch again.

After all the above switches are stated on the command line, THE treats the remainder of the command line as files or directories to edit.

PROFILE FILE

Various session defaults may be changed on startup for an individual by using a profile file. This file contains various commands that set the current environment, including key bindings. This profile file can also be used to process commands in batch mode.

THE will always execute a default profile. Appendix 1 defines the name of the default profile on different platforms.

An example of a profile might be to change all occurrences of *alligator* to *crocodile* in the file *file.ext* in batch mode, a profile file; *prf.prf* with the following commands would be used:

```
'c/alligator/crocodile/ * *'
'file'
```

and the command

```
the −b −p prf.prf file.ext
```

would be issued.

This changes the first string enclosed in delimiters (generally any non−alphabetic character not in the string itself) to the second string for every line (*) starting at the current line (0 initially) changing each occurrence on a line (*).

Maybe you only want to change a string after the first line that contains the string *donkey* , but only change the second occurrence of that string. The profile commands would then be:

```
'/donkey/'
'c/alligator/crocodile/ * 1 2'
'file'
```

The change command uses a target specification as its first parameter after the string details. A target can be a number of lines, an absolute line number, BLANK, ALL or a string.

Number of line targets consist of either a positive integer, for referencing lines toward the end of the file, negative for referencing toward the start of the file or * , which is all the remaining lines in the file or −* which is all lines toward the start of the file.

An absolute line number in the form of *:n* is the line number in a file, starting with line number 1.

The HTML version of this manual was inspired by Judith Grobe Sachs

Generated on: 2 Nov 2002

Return to Table of Contents

COMMAND REFERENCE

ADD − add blank line

Syntax:

Add *[n]*

Description:

The ADD command inserts *n* blank lines after the current line , if issued from the command line or after the focus line , if issued from the filearea or prefix area .

If SET NEWLINES is set to ALIGNED, the cursor is positioned in the column corresponding to the first column not containing a space in the line above.

If SET NEWLINES is set to LEFT, the cursor is positioned in the first column.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1

See Also:

SOS ADDLINE

Status:

Complete

ALERT − display a user configurable dialog box with notification

Syntax:

ALERT */prompt/ [EDITfield [/val/]] [TITLE /title/] [OK|OKCANCEL|YESNO|YESNOCANCEL] [DEFBUTTON n]*

Description:

The ALERT command is identical to the DIALOG command except that if SET BEEP is on, a beep is played.

On exit from the ALERT command, the following Rexx variables are set:

ALERT.0 − 2

ALERT.1 − value of 'EDITfield'

ALERT.2 − button selected as specified in the call to the command.

The colours for the alert box are the same as for a dialog box, except the prompt area which uses the colour set by SET COLOR ALERT.

Compatibility:

XEDIT: N/A

KEDIT: Compatible. Does not support bitmap icons or font options.

See Also:

POPUP, DIALOG, READV, SET COLOR

Status:

Complete.

ALL − select and display restricted set of lines

Syntax:

ALL *[rtarget]*

Description:

The ALL command allows for the selective display, and editting (subject to SET SCOPE) of lines that match the specified target. This target consists of any number of individual targets seperated by '' (logical and) or '|' (logical or).

For example, to display all lines in a file that contain the strings 'ball' and 'cat' on the same line or the named lines .fred or .bill, use the following command

ALL /ball/ &/cat/ | .fred | .bill

Logical operators act left to right, with no precedence for .

rtarget can also be specified as a regular expression. The syntax of this is "Regexp /re/". eg ALL R /[0−9].*$/

ALL without any arguments, is the equivalent of setting the selection level of all lines in your file to 0 and running SET DISPLAY 0 0.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SET SCOPE, SET DISPLAY, SET SELECT

Status:

Complete.

BACKWARD − scroll backwards [n] screens

Syntax:

BAckward *[n|*]*

Description:

The BACKWARD command scrolls the file contents backwards through the file *n* or * screens.

If 0 is specified as the number of screens to scroll, the last line of the file becomes the current line .

If the BACKWARD command is issued while the current line is the Top−of−File line , the last line of the file becomes the current line .

Compatibility:

XEDIT: Compatible.

KEDIT: Does not support HALF or Lines options.

Default:

1

See Also:

FORWARD, TOP

Status:

Complete

BOTTOM − move to the bottom of the file

Syntax:

Bottom

Description:

The BOTTOM command moves to the very end of the current file. The last line of the file is set to the current line .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

FORWARD, TOP

Status:

Complete

CANCEL − quit from all unaltered files in the ring

Syntax:

CANcel

Description:

The CANCEL command exits from THE quickly by executing a QQUIT command for every file in the ring that does not have any outstanding alterations.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

CCANCEL

Status:

Complete.

CAPPEND − append text after column pointer

Syntax:

CAppend *[text]*

Description:

The CAPPEND command moves the column pointer to the end of the focus line and appends the specified *text* .

If no *text* is specified, the column pointer moves to the first trailing space.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

CLAST

Status:

Complete.

CCANCEL − qquit from all files in the ring

Syntax:

CCancel

Description:

The CCANCEL command exits from THE quickly by executing the QQUIT command for every file in the ring. Any changes made to any of the files since the last SAVE will be lost.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

CANCEL

Status:

Complete.

CDELETE − delete text starting at column pointer

Syntax:

CDelete *[column target]*

Description:

The CDELETE command deletes characters starting from the current column pointer for the specified column target .

If no column target is specified, the character at the column pointer is deleted.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Status:

Incomplete. No string targets.

CFIRST − move column pointer to beginning of zone

Syntax:

CFirst

Description:

The CFIRST command moves the column pointer to the beginning of the zone.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SET ZONE

Status:

Complete.

CHANGE − change one string to another

Syntax:

Change *[/string1/string2/ [target] [n] [m]]*

Description:

The CHANGE command changes one string of text to another.

The first parameter to the change command is the old and new string values, seperated by delimiters. The first non alphabetic character after the 'change' command is the delimiter.

target specifies how many lines are to be searched for occurrences of *string1* to be changed.

n determines how many occurrences of *string1* are to be changed to *string2* on each line. *n* may be specified as '*' which will result in all occurrences of *string1* will be changed. '*' is equivalent to the current WIDTH of the line.

m determines from which occurrence of *string1* on the line changes are to commence.

If no arguments are supplied to the CHANGE command, the last change command, if any, is re−executed.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1 1 1

See Also:

SCHANGE

Status:

Complete.

CINSERT − insert text starting at the column pointer

Syntax:

CInsert *text*

Description:

The CINSERT command inserts *text* starting at the column position.

text can include leading or trailing space characters. Thus CINSERT immediatley followed by 5 spaces, will insert 4 space characters. The first space character is the command seperator.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Status:

Complete.

CLAST − move the column pointer to end of zone

Syntax:

CLAst

Description:

The CLAST command moves the column pointer to the end of the zone.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SET ZONE

Status:

Complete.

CLOCATE − move the column pointer

Syntax:

CLocate *column target*

Description:

The CLOCATE command scans the file for the specified *column target* beginning with the column following (or preceding) the column pointer.

Column targets can be specified as absolute targets, relative targets or string targets.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Status:

Incomplete. No string targets.

CMATCH − find matching bracket character

Syntax:

CMATCH

Description:

The CMATCH command searches for the matching bracket character to the character under the cursor.

It handles nested sets of matching pairs. The matching character pairs are []{}<>().

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Status:

Complete.

CMSG − display text on command line

Syntax:

CMSG *[text]*

Description:

The CMSG command, primarily used in macros, displays *text* on the command line.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

EMSG, MSG

Status:

Complete.

COMMAND − execute a command without translation

Syntax:

COMMAND *command [options]*

Description:

The COMMAND command executes the specified *command* without synonym or macro translation. THE does not attempt to execute the command as a macro even if SET IMPMACRO is ON. The command will be passed to the operating system if SET IMPOS is ON.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

COMPRESS − reduce spaces to tabs

Syntax:

COMPress *[target]*

Description:

The COMPRESS command reduces multiple occurrences of spaces and replaces them with tab characters in the *target* lines. The current tab columns (set by SET TABS) are used in determining where tab characters will replaces spaces.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

EXPAND, SET TABS

Status:

Complete.

CONTROLCHAR − allow control characters to be entered

Syntax:

CONTROLChar

Description:

The CONTROLCHAR command prompts the user to enter a control character; an ASCII character between 1 and 31 inclusive.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Status:

Complete.

COPY − copies text from one position to another

Syntax:

COPY *target1 target2*
COPY *BLOCK [RESET]*

Description:

With the first form of the COPY command, text is copied from *target1* to the line specified by *target2* . Text can only be copied within the same view of the file.

The second form of the COPY command copies text within the currently marked block to the current cursor position. The text can be in the same file or a different file.

Compatibility:

XEDIT: COPY BLOCK not available.
KEDIT: Adds extra functionality with [RESET] option.
With the cursor in the marked block this command in KEDIT
acts like DUPLICATE BLOCK.

Status:

Complete.

COVERLAY − overlay text starting at the column pointer

Syntax:

COVerlay *text*

Description:

The COVERLAY command overlays the supplied *text* onto the characters following the column position.

Spaces in the *text* do not destroy the existing characters.

An underscore character "_" in the *text* places a space in the corresponding character position.

Therefore you cannot use the COVERLAY command to place underscores in a line.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

CREPLACE − replace text starting at the column pointer

Syntax:

CReplace *text*

Description:

The CREPLACE command replaces the current characters after the column pointer with the supplied *text* .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

CURSOR − move cursor to specified position

Syntax:

CURsor *Column*
CURsor *Screen UP|DOWN|LEFT|RIGHT*
CURsor *Screen row [col]*
CURsor *[Escreen] UP|DOWN*
CURsor *[Escreen|Kedit] LEFT|RIGHT*
CURsor *[Escreen] row [col]*
CURsor *CUA UP|DOWN|LEFT|RIGHT*
CURsor *CMdline [n]*
CURsor *HOME [SAVE]*
CURsor *File line [col]*
CURsor *GOTO line col*
CURsor *Mouse*

Description:

The CURSOR command allows the user to specify where the cursor is to be positioned.

CURSOR *Column* moves the cursor to the current column of the focus line .

CURSOR *Screen UP | DOWN | LEFT | RIGHT* moves the cursor in the indicated direction one line or column. If the cursor is positioned on the first or last line of the screen, the cursor wraps to the first or last enterable line. If the cursor is positioned on the left or right edges of the screen, the cursor moves to the left or right edge of the screen on the same line.

CURSOR *Screen row* [*col*] is similar to CURSOR *Escreen row* [*col*], but all coordinates are relative the the top left corner of the screen, not the top left corner of the filearea . Hence, 1,1 would be an invalid cursor position because it would result in the cursor being moved to the idline . Specification of *row* and/or *col* outside the boundaries of the logical window is regarded as an error.

CURSOR [*Escreen*] *UP | DOWN | LEFT | RIGHT* is similar to CURSOR *Screen UP | DOWN | LEFT | RIGHT* , except that where scrolling of the window is possible, then scrolling will take place.

CURSOR [*Escreen*] *row* [*col*] moves the cursor to the specified *row* / *col* position within the filearea . The top left corner of the filearea is 1,1. *row* and *col* may be specified as '=' , which will default to the current row and/or column position. If *row* or *col* are greater than the maximum number of rows or columns in the filearea , the cursor will move to the last row/column available. If the specified *row* is a reserved line , scale line or tab line an error will be displayed. If the *row* specified is above the Top−of−File line or below the Bottom−of−File line the cursor will be placed on the closest one of these lines.

CURSOR *Kedit LEFT | RIGHT* mimics the default behaviour of CURL and CURR in KEDIT.

CURSOR *CUA UP | DOWN | LEFT | RIGHT* moves the cursor in the indicated direction one line or column. The behaviour of the cursor at the the end of a line and at the start of a line is consistent with the Common User Access (CUA) definition.

CURSOR *CMdline* moves the cursor to the indicated column of the command line .

CURSOR *HOME* moves the cursor to the first column of the command line (if not on the command line), or to the last row/column of the filearea if on the command line. With the [*SAVE*] option, the cursor will move to the last row/column of the filearea or prefix area (which ever was the last position) if on the command line .

CURSOR *File* moves the cursor to the line and column of the file. If the line and/or column are not currently displayed, an error message is displayed.

CURSOR *GOTO* moves the cursor to the specified line and column of the file, whether the row and column are currently displayed or not. If the *line* and *col* are currently displayed, then this command behaves just like CURSOR *FIle* . If not, then the current line will be changed to the specified line .

CURSOR *Mouse* moves the cursor to the position where a mouse button was last activated. This command is specific to THE.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible. Added GOTO option.

Status:

Complete.

DEFINE − assign one or many commands to a key or mouse event

Syntax:

DEFine *key−name [REXX] [command [args] [[#command [args]...]]]*

DEFine *mouse−key−definition IN window [REXX] [command [args] [[#command [args]...]]]*

Description:

The DEFINE command allows the user to assign one or many commands and optional parameter(s) to a key or a mouse button specification.

Commands may be abbreviated.

If multiple commands are assigned, then the LINEND setting must be ON and the LINEND character must match the character that delimits the commands at the time that the DEFINE command is executed. LINEND can be OFF at the time the key is pressed.

With no arguments, any existing definition for that key is removed and the key reverts back to its default assignation (if it had any).

key−name corresponds to the key name shown with the SHOWKEY command.

If the optional keyword; *REXX* , is supplied, the remainder of the command line is treated as a Rexx macro and is passed onto the Rexx interpreter (if you have one) for execution.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

KEDIT does not allow multiple commands except as KEXX
macros.

See Also:

SHOWKEY, SET LINEND

Status:

Complete.

DELETE − delete lines from a file

Syntax:

DELete *[target]*

Description:

The DELETE command removes lines from the current file. The number of lines removed depends on the *target* specified. Lines are removed starting with the focus line .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1

See Also:

SOS DELLINE

Status:

Complete.

DIALOG − display a user configurable dialog box

Syntax:

DIALOG */prompt/ [EDITfield [/val/]] [TITLE /title/] [OK|OKCANCEL|YESNO|YESNOCANCEL] [DEFBUTTON n]*

Description:

The DIALOG command displays a dialog box in the middle of the screen with user−configurable settings.

The mandatory *prompt* parameter, is the text of a prompt displayed near the top of the dialog window. Up to 10 lines can be displayed by separating lines with a character (decimal 10).

EDITfield creates a user enterable field, with a default value of *val* , if supplied. While the cursor is in the editfield, "normal" edit keys are in effect. See READV for more details on keys that are useable in the editfield. The same keys that exit from the READV command also exit the editfield. On exit from the editfield, the first button becomes active.

title specifies optional text to be displayed on the border of the dialog box.

The type of button combination can be specifed as one of the following:

OK − just an OK button is displayed

OKCANCEL − an OK and a CANCEL button are displayed

YESNO − a YES and a NO button are displayed

YESNOCANCEL − a YES, a NO and a CANCEL button are displayed

If no button combination is selected, an OK button is displayed.

If *DEFBUTTON* is specified, it indicates which of the buttons is to be set as the active button. This is a number between 1 and the number of buttons displayed. By default, button 1 is active. If *EDITfield* is specified, no active button is set.

The active button can be selected by pressing the TAB key; to exit from the DIALOG, press the RETURN or ENTER key, or click the first mouse button on the required button.

On exit from the DIALOG command, the following Rexx variables are set:

DIALOG.0 − 2

DIALOG.1 − value of 'EDITfield'

DIALOG.2 − button selected as specified in the call to the command.

The colours used for the dialog box are:

Border − SET COLOR DIVIDER

Prompt area − SET COLOR DIALOG

Editfield − SET COLOR CMDLINE

Inactive button − SET COLOR BLOCK

Active button − SET COLOR CBLOCK

Compatibility:

XEDIT: N/A

KEDIT: Compatible. Does not support bitmap icons or font options.

See Also:

POPUP, ALERT, READV, SET COLOR

Status:

Complete.

DIRECTORY − list the specified directory as an editable file

Syntax:

DIRectory *[file specification]*

Description:

The DIRECTORY command displays all files matching the specified *file specification* .

When no parameter is supplied, all files in the current directory are displayed subject to any SET DIRINCLUDE restrictions.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

LS, SET DIRINCLUDE

Status:

Complete.

DOS − execute an operating system command

Syntax:

DOS *[command]*

Description:

The DOS command executes the supplied operating system *command* or runs an interactive shell if no *command* is supplied.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

OS, !

Status:

Complete.

DOSNOWAIT − execute an operating system command − no prompt

Syntax:

DOSNowait *command*

Description:

The DOSNOWAIT command executes the supplied operating system command not waiting for the user to be prompted once the command has completed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

OSNOWAIT

Status:

Complete.

DOSQUIET − execute an operating system command quietly

Syntax:

DOSQuiet *command*

Description:

The DOSQUIET command executes the supplied operating system *command* as quietly as possible.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

OSQUIET

Status:

Complete.

DOWN − move forward in the file a number of lines

Syntax:

Down *[relative target]*

Description:

The DOWN command moves the current line forwards the number of lines specified by the *relative target* . This *relative target* can only be a positive integer or the character "*".

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

1

See Also:

NEXT, UP

Status:

Complete.

DUPLICATE − duplicate lines

Syntax:

DUPlicate *[n [target|BLOCK]]*

Description:

The DUPLICATE command copies the number of lines extrapolated from *target* or the marked *BLOCK* , *n* times.

Compatibility:

XEDIT: Equivalent of DUPLICAT command.
KEDIT: Compatible.

Status:

Complete.

EDIT − edit another file or switch to next file

Syntax:

Edit *[file]*

Description:

The EDIT command allows the user to edit another *file* . The new file is placed in the file ring . The previous file being edited remains in memory and can be returned to by issuing an EDIT command without any parameters. Several files can be edited at once, and all files are arranged in a ring, with subsequent EDIT commands moving through the ring, one file at a time.

Compatibility:

XEDIT: Does not provide options switches.
KEDIT: Does not provide options switches.

See Also:

THE, XEDIT, KEDIT

Status:

Complete.

EDITV − set and retrieve persistent macro variables

Syntax:

EDITV *GET|PUT|GETF|PUTF var1 [var2 ...]*
EDITV *SET|SETF var1 value1 [var2 value2 ...]*
EDITV *SETL|SETLF|SETFL var1 value1*
EDITV *LIST|LISTF [var1 ...]*

Description:

The EDITV command manipulates variables for the lifetime of the edit session or the file, depending on the subcommand used.
Edit variables are useful for maintaining variable values from one execution of a macro to another.
EDITV GET, PUT, GETF and PUTF are only valid from within a macro as they reference Rexx variables. All other subcommands are valid from within a macro or from the command line.
EDITV GET sets a Rexx macro variable, with the same name as the edit variable, to the value of the edit variable.
EDITV PUT stores the value of a Rexx macro variable as an edit variable.

EDITV SET stores an edit variable with a value.

EDITV SET can only work with variable values comprising a single space−seperated word. To specify a variable value that contains spaces, use EDITV SETL.

EDITV LIST displays the values of the specified edit variables, or all variables if no edit variables are specified.

EDITV GETF, PUTF, SETF, SETLF, SETFL, and LISTF all work the same way as their counterparts without the F, but the variables are only available while the particular file is the current file. This enables you to use the same edit variable name but with different values for different files.

Compatibility:

XEDIT: N/A

KEDIT: Compatible

Status:

Complete.

EMSG − display message

Syntax:

EMSG *[message]*

Description:

The EMSG command displays an *message* on the message line . This command is usually issued from a macro file.

Compatibility:

XEDIT: Does not support [mmmnnns text] option

KEDIT: Compatible

See Also:

CMSG, MSG

Status:

Complete.

ENTER − execute a command

Syntax:

enter *[CUA]*

Description:

The ENTER command executes the command currently displayed on the command line, if the cursor is currently displayed there. If the key associated with ENTER is pressed while in the filearea , then the cursor will move to the first column of the next line. If the cursor is in the prefix area , any pending prefix commands will be executed. If the mode is currently in 'insert' , then a new line is added and the cursor placed on the next line depending on the value of SET NEWLINES .

This command can only be used by assigning it to a function key with the DEFINE command.

With the optional *CUA* argument, when in the FILEAREA , the enter command acts like the SPLIT command.

This command will be removed in a future version.

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

SOS EXECUTE, ENTER

Status:

Complete.

EXPAND − expand tab characters to spaces

Syntax:

EXPand *[target]*

Description:

The EXPAND command converts all tab characters to spaces in the *target* depending on the size of a tab determined by the SET TABS command.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

COMPRESS, SET TABS

Status:

Complete.

EXTRACT − obtain various internal information about THE

Syntax:

EXTract */item/[...]*

Description:

The EXTRACT command is used to relay information about settings within THE from within a Rexx macro. EXTRACT is only valid within a Rexx macro.

The '/' in the syntax clause represents any delimiter character.

For a complete list of *item* s that can be extracted, see the section; QUERY, EXTRACT and STATUS .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

FFILE − force a FILE of the current file to disk

Syntax:

FFile *[filename]*

Description:

The FFILE command writes the current file to disk to the current file name or to the supplied *filename* . Unlike the FILE command, if the optional *filename* exists, this command will overwrite the file.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

With no parameters, the current file is written.

See Also:

FILE, SAVE, SSAVE

Status:

Complete

FILE − write the current file to disk and remove from ring

Syntax:

FILE *[filename]*

Description:

The FILE command writes the current file to disk to the current file name or to the supplied *filename* . Unlike the FFILE command, if the optional *filename* exists, this command will not overwrite the file.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

With no parameters, the current file is written.

See Also:

FFILE, SAVE, SSAVE

Status:

Complete

FILLBOX − fill the marked block with a character

Syntax:

FILLbox *[c]*

Description:

The FILLBOX command fills the marked block with the specified character, *c* . If no parameters are supplied and the command is run from the command line, then the block will be filled with spaces. If the command is not run from the command line, the user is prompted for a character to fill the box.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete

FIND − locate forwards the line which begins with the supplied string

Syntax:

Find *[string]*

Description:

The FIND command attempts to locate a line towards the end of the file that begins with *string* . If the optional *string* is not supplied the last *string* used in any of the family of find commands is used.
string can contain two special characters:
space − this will match any single character in the target line
underscore − this will match any single space in the target line

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FINDUP, NFIND, NFINDUP

Status:

Complete

FINDUP − locate backwards the line which begins with the supplied string

Syntax:

FINDUp *[string]*

Description:

The FINDUP command attempts to locate a line towards the start of the file that begins with *string* . If the optional *string* is not supplied the last *string* used in any of the family of find commands is used.
string can contain two special characters:

space − this will match any single character in the target line
underscore − this will match any single space in the target line

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FIND, NFIND, NFINDUP, FUP

Status:

Complete

FORWARD − scroll forwards [n] screens

Syntax:

FOrward *[n]*

Description:

The FORWARD command scrolls the file contents forwards *n* screens.

If 0 is specified as the number of screens to scroll, the Top−of−File line becomes the current line .

If the FORWARD command is issued while the current line is the Bottom−of−File line , the Top−of−File line becomes the current line .

Compatibility:

XEDIT: Compatible.
KEDIT: Does not support HALF or Lines options.

Default:

1

See Also:

BACKWARD, TOP

Status:

Complete

FUP − locate backwards the line which begins with the supplied string

Syntax:

FUp *[string]*

Description:

The FUP command is a synonym for the FINDUP command.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FIND, NFIND, NFINDUP, FINDUP

Status:

Complete

GET − insert into file the contents of specified file

Syntax:

GET *[filename] [fromline] [numlines]*
GET *CLIP: [STREAM|BOX|LINE]*

Description:

The GET command reads a file into the current file, inserting lines after the current line.

When no *filename* is supplied the temporary file generated by the last PUT or PUTD command is used.

When *fromline* is specified, reading of the file begins at the line number specified. If *fromline* is not specifed, reading begins at line 1.

When *numlines* is specified, reading of the file ends when the specified number of lines has been read. If *numlines* is not specified, or *numlines* is specified as '*' , all files from the *fromline* to the end of file are read.

The second form implements interaction with the system clipboard. The optional parameter indicates how the contents of the clipboard is to be inserted into the file. If not supplied the contents of the clipboard is treated as a LINE block. This option only available for X11, OS/2 and Win32 ports of THE. (Incomplete)

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

PUT, PUTD

Status:

Complete

HELP − edit help file for THE

Syntax:

HELP

Description:

The HELP command displays help for the editor. Uses THE_HELP_FILE environment variable to point to the help file. See Appendix 1 for details on this and other environemnt variables.

Compatibility:

XEDIT: Similar in concept.
KEDIT: Similar in concept.

Status:

Complete.

HIT − simulate hitting of the named key

Syntax:

HIT *key*

Description:

The HIT command enables the simulation of hitting the named *key* . This is most useful from within a macro.

Be very careful when using the HIT command with the DEFINE command. If you assign the HIT command to a key, DO NOT use the same key name. eg. DEFINE F1 HIT F1 This will result in an infinite processing loop.

Compatibility:

XEDIT: N/A
KEDIT: Similar, but more like the MACRO command.

Status:

Complete.

INPUT − insert the command line contents into the file

Syntax:

Input *[string]*

Description:

The INPUT command inserts the *string* specified on the command line into the current file after the current line .

If SET INPUTMODE FULL is in effect, and the INPUT command is entered on the command line with no arguments, THE is put into full input mode. If the prefix area is on, it is turned off, the cursor moved to the filearea and blank lines inserted into the file from the current line to the end of the screen.

To get out of full input mode, press the key assigned to the CURSOR HOME [SAVE] command.

Compatibility:

XEDIT: Does not provide full input mode option.

KEDIT: Does not provide full input mode option.

Status:

Complete. Except for full input mode capability.

JOIN − join a line with the line following

Syntax:

Join *[ALigned] [Column|CURSOR]*

Description:

The JOIN command makes one line out of the focus line and the line following.

If *Aligned* is specified, any leading spaces in the following line are ignored. If *Aligned* is not specified, all characters, including spaces are added.

If *Column* (the default) is specified, the current line is joined at the current column location.

If *CURSOR* is specified, the focus line is joined at the cursor position.

Compatibility:

XEDIT: Compatible.

Does not support Colno option

KEDIT: Compatible.

See Also:

SPLIT, SPLTJOIN

Status:

Complete.

KEDIT − edit another file or switch to next file

Syntax:

Kedit *[file]*

Description:

The KEDIT command allows the user to edit another *file* . The new file is placed in the file ring . The previous file being edited remains in memory and can be returned to by issuing a KEDIT command without any parameters. Several files can be edited at once, and all files are arranged in a ring, with subsequent KEDIT commands moving through the ring, one file at a time.

Compatibility:

XEDIT: Does not provide options switches.

KEDIT: Does not provide options switches.

See Also:

EDIT, THE, XEDIT

Status:

Complete.

LEFT − scroll the screen to the left

Syntax:

LEft *[n|HALF]*

Description:

The LEFT command scrolls the screen to the left.

If *n* is supplied, the screen scrolls by that many columns.

If *HALF* is specified the screen is scrolled by half the number of columns in the filearea .
If 'FULL' is specified the screen is scrolled by the number of columns in the filearea .
If no parameter is supplied, the screen is scrolled by one column.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

RIGHT, RGTLEFT

Status:

Complete.

LOCATE − search for a target

Syntax:

[Locate] *target [command]*

Description:

The LOCATE command searches for the next or previous occurrence of the specified *target* . If no parameter is supplied, LOCATE uses the the last target specified. If no prior target has been specified, an error message is displayed.

target can also be specified as a regular expression. The syntax of this is "Regexp /re/". eg LOCATE RE /[0−9].*$/

With an optional *command* , this command is executed after finding the *target* .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

LOWERCASE − change uppercase characters to lowercase

Syntax:

LOWercase *[target]*

Description:

The LOWERCASE command changes all uppercase characters in all lines up to the *target* line to lowercase. All other characters remain untouched.

Compatibility:

XEDIT: Equivalent of LOWERCAS command.
KEDIT: Compatible.

See Also:

UPPERCASE

Status:

Complete.

LS − list the specified directory as an editable file

Syntax:

LS *[file specification]*

Description:

The LS command displays all files matching the specified *file specification* .

When no parameter is supplied, all files in the current directory are displayed subject to any SET DIRINCLUDE restrictions.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

DIRECTORY, SET DIRINCLUDE

Status:

Complete.

MACRO − execute a macro command file

Syntax:

MACRO *[?] filename [arguments ...]*

Description:

The MACRO command executes the contents of the specified *filename* as command line commands. The *filename* can contain either a series of THE commands, or can be a Rexx program. The *filename* is considered a macro .

Rexx macros can be passed optional *arguments* .

With the optional *?* parameter, interactive tracing of the Rexx macro is possible, but this does not set interactive tracing on;

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

MARK − mark a portion of text

Syntax:

MARK *Box [line1 col1 line2 col2]*
MARK *Line [line1 line2]*
MARK *Stream [line1 col1 line2 col2]*
MARK *Column [col1 col2]*
MARK *Word [line1 col1]*
MARK *CUA [LEFT|RIGHT|UP|DOWN|START|END|FOrward|BAckward|TOP|Bottom|MOUSE]*

Description:

The MARK command marks a portion of text for later processing by a COPY , MOVE or DELETE command. This marked area is known as a block .

When the MARK command is executed with the optional line/column arguments, these values are used to specify the position of the marked block . Without the optional arguments, the position of the cursor is used to determine which portion of text is marked.

line1 and *line2* specify the first or last line of the marked block.

col1 and *col2* specify the first or last column of the marked block.

Any existing marked block will be replaced with the block specified in this command.

When marking a word block , *line1* and *col1* refer to any position within the word.

Compatibility:

XEDIT: N/A

KEDIT: Adds CUA, WORD, and COLUMN options and position specifiers.

Status:

Complete.

MODIFY − display current SET command for alteration

Syntax:

MODify *set−command*

Description:

The MODIFY command displays the current setting of a SET command on the command line enabling the user to change that setting.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SET, QUERY

Status:

Complete.

MOVE − move a portion of text

Syntax:

MOVE *target1 target2*

MOVE *BLOCK [RESET]*

Description:

The MOVE command copies the contents of a portion of the file to the same or a different file, and deletes the marked portion from the original file.

The first form of the MOVE command, moves the portion of the file specified by *target1* to the line spcified by *target2* in the same file.

The second form of the MOVE command moves the contents of the marked block to the current cursor position. If the optional [*RESET*] argument is supplied, the marked block is reset as though a RESET BLOCK command had been issued.

Compatibility:

XEDIT: N/A

KEDIT: Adds extra functionality with [RESET] option.

Status:

Incomplete. First form is not supported.

MSG − display message on error line

Syntax:

MSG *[message]*

Description:

The MSG command displays a *message* on the message line . This command is usually issued from a macro file. This is similar to EMSG , but MSG does not sound the bell if SET BEEP is on.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

CMSG, EMSG

Status:

Complete.

NEXT − move forward in the file a number of lines

Syntax:

Next *[relative target]*

Description:

The NEXT command moves the current line forwards the number of lines specified by the relative target . This relative target can only be a positive integer or the character "*".

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1

See Also:

DOWN, UP

Status:

Complete.

NEXTWINDOW − switch focus of editing session to another file

Syntax:

NEXTWindow

Description:

The NEXTWINDOW command moves the focus of the editing session to the other screen (if SET SCREEN 2 is in effect) or to the next file in the ring .

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

PREVWINDOW, EDIT, SET SCREEN

Status:

Complete.

NFIND − locate forwards the line which does NOT begin with the supplied string

Syntax:

NFind *[string]*

Description:

The NFIND command attempts to locate a line towards the end of the file that does NOT begin with *string* . If the optional *string* is not supplied the last *string* used in any of the family of find commands is used.

string can contain two special characters:
space − this will match any single character in the target line
underscore − this will match any single space in the target line

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FIND, FINDUP, NFINDUP

Status:

Complete

NFINDUP − locate backwards the line which does NOT begin with the supplied string

Syntax:

NFINDUp *[string]*

Description:

The NFINDUP command attempts to locate a line towards the start of the file that does NOT begin with *string* . If the optional *string* is not supplied the last *string* used in any of the family of find commands is used.

string can contain two special characters:

space − this will match any single character in the target line
underscore − this will match any single space in the target line

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FIND, FINDUP, NFIND, NFUP

Status:

Complete

NFUP − locate backwards the line which does NOT begin with the supplied string

Syntax:

NFUp *[string]*

Description:

The NFUP command is a synonym for the NFINDUP command.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

FIND, FINDUP, NFIND, NFINDUP

Status:

Complete

NOMSG − execute a command suppressing any messages

Syntax:

NOMSG *command [arguments]*

Description:

The NOMSG command executes the supplied *command* but suppresses messages that would normally be displayed as a result of the command.

Optional *arguments* may be passed to the *command* .

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete.

NOP − no operation command

Syntax:

NOP

Description:

The NOP command does nothing. It is used as a means of turning off an assignment to a key.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

DEFINE

Status:

Complete.

OS − execute an operating system command

Syntax:

OS *[command]*

Description:

The OS command executes the supplied operating system *command* or runs an interactive shell if no *command* is supplied.

Compatibility:

XEDIT: N/A

KEDIT: Equivalent to DOS command.

See Also:

DOS, !

Status:

Complete.

OSNOWAIT − execute an operating system command − no prompt

Syntax:

OSNowait *command*

Description:

The OSNOWAIT command executes the supplied operating system *command* not waiting for the user to be prompted once the command has completed.

Compatibility:

XEDIT: N/A

KEDIT: Equivalent of DOSNOWAIT command.

See Also:

DOSNOWAIT

Status:

Complete.

OSQUIET − execute an operating system command quietly

Syntax:

OSQuiet *command*

Description:

The OSQUIET command executes the supplied operating system *command* as quietly as possible.

Compatibility:

XEDIT: N/A

KEDIT: Equivalent of DOSQUIET command.

See Also:

DOSQUIET

Status:

Complete.

OSREDIR − execute an operating system command and capture output

Syntax:

OSRedir *filename command [arguments ...]*

Description:

The OSREDIR command executes the supplied operating system *command* and redirects output destined for STDOUT and STDERR to the specified *filename* .

Optional *arguments* may be supplied to the *command* .

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

OS, OSQUIET, OSNOWAIT

Status:

Complete.

OVERLAYBOX − overlay marked block on current cursor position

Syntax:

OVERLAYBox

Description:

The OVERLAYBOX command overlays the contents of the marked block ; box block or line block , over the characters or lines at the focus line and focus column .

This command implies that only box block s are handled. This used to be the case, and for compatibility reasons the name remains.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

MOVE, COPY

Status:

Complete.

POPUP − display popup menu

Syntax:

POPUP *[MOUSE|TEXT|CENTER|CENTRE|BELOW|ABOVE] [INITIAL n] [ESCAPE keyname] /item1[/item2/...]*

Description:

The POPUP command allows the user to create and display a popup menu containing a list of selectable options.

The location of the popup menu is specified by the first parameter.

MOUSE specifies that the top left corner of the popup menu is to be displayed where the mouse cursor currenlty is displayed. This option is only valid if the popup window is initiated from a macro assigned to a mouse event.

TEXT specifies that the top left corner of the popup menu is to be displayed where the text cursor is displayed.

ABOVE specifies that the bottom row of the popup window is to be displayed above the line where the text cursor is displayed. The popup window will use at most from the line above the text cursor to the top of the screen.

BELOW specifies that the top row of the popup window is to be displayed below the line where the text cursor is displayed. The popup window will use at most from the line below the text cursor to the bottom of the screen.

CENTRE or *CENTER* specifies that the popup window is centred in the middle of the screen. This option will use all of the screen to display the popup window if necessary.

If the location is not specified, then the default is *CENTRE*

INITIAL specifies the item to be highlighted when the popup window is first displayed. This value must be within the bounds of the items specified.

ESCAPE specifies the keyname that can be used to quit from the popup window without making a selection. By default 'q' will quit. Only keynames that are valid with the DEFINE command are allowed.

On return from the popup menu, the following Rexx variables are set:

popup.0 = 2

popup.1 = Item selected or empty string if no item selected.

popup.2 = Item number selected or zero if no item selected.

popup.3 = Item number on which the cursor was last positioned.

If mouse support is available, an item is selectable by clicking the first mouse button on the item. To quit from the popup window without making a selection, click the mouse outside the popup window, or on the border of the window.

Keyboard keys that take effect in the POPUP command are CURU, CURD, CURL, CURR, and ENTER.

The colours used for the popup are:

Border − SET COLOR DIVIDER

Non−current line − SET COLOR BLOCK

Current line − SET COLOR CBLOCK

Compatibility:

XEDIT: N/A

KEDIT: KEDIT does not support INITIAL, ESCAPE, ABOVE or BELOW options.

See Also:

DIALOG, ALERT

Status:

Complete.

PRESERVE − save various editor settings

Syntax:

PREServe

Description:

The PRESERVE command saves various editing settings at the time the command is issued. These settings can then be restored by using the RESTORE command.

The following view level settings are saved: ARBCHAR ARROW CASE CMDLINE CURLINE DISPLAY HEX HEXSHOW HIGHLIGHT IDLINE IMPMACRO IMPOS INPUTMODE LINEND MACRO MARGINS MSGLINE MSGMODE NEWLINE NUMBER POSITION PREFIX SCALE SCOPE SHADOW STAY SYNONYM TABLINE TABS VERIFY VERSHIFT WORD WORDWRAP ZONE

The following file level settings are saved: AUTOSAVE BACKUP COLOUR ECOLOUR EOLOUT TABSOUT

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

RESTORE

Status:

Complete.

PREVWINDOW − switch focus of editing session to another file

Syntax:

PREVWindow

Description:

The PREVWINDOW command moves the focus of the editing session to the other screen (if SET SCREEN 2 is in effect) or to the previous file in the ring .

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

NEXTWINDOW, EDIT, SET SCREEN

Status:

Complete.

PRINT − send text to default printer or print spooler

Syntax:

PRint *[target] [n]*
PRint *LINE [text]*
PRint *STRING [text]*
PRint *FORMfeed*
PRint *CLOSE*

Description:

The PRINT command writes a portion of the current file to the default printer or print spooler, or text entered on the command line.

PRINT [*target*] [*n*] Sends text from the file contents up to the *target* to the printer followed by a CR/LF (DOS) or LF(UNIX) after each line. When [*n*] is specified, this sends a formfeed after [*n*] successive lines of text. PRINT *LINE* [*text*] Sends the remainder of the *text* on the command line to the printer followed by a LF(UNIX), CR(MAC) or CR/LF (DOS). PRINT *STRING* [*text*] Sends the remainder of the *text* on the command line to the printer without any trailing line terminator. PRINT *FORMfeed* Sends a formfeed (^L) character to the printer. PRINT *CLOSE* Closes the printer spooler.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SET PRINTER

Status:

Complete.

PUT − write part of a file to another

Syntax:

PUT *[target] [filename]*

Description:

The PUT command writes a portion of the current file, defined by *target* to another file, either explicit or temporary.

When no *filename* is supplied the temporary file used for PUT and GET commands is overwritten.

When a *filename* is supplied the portion of the file written out is appended to the specified file.

If 'CLIP:' is used in place of *filename* , the portion of the file specified by *target* is written to the clipboard. This option only available for X11, OS/2 and Win32 ports of THE.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

PUTD, GET

Status:

Complete.

PUTD − write part of a file to another and delete

Syntax:

PUTD *[target] [filename]*

Description:

The PUTD command writes a portion of the current file, defined by *target* to another file, either explicit or temporary, and then deletes the lines written.

When no *filename* is supplied the temporary file used for PUT and GET commands is overwritten.
When a *filename* is supplied the portion of the file written out is appended to the specified file.
If 'CLIP:' is used in place of *filename* , the portion of the file specified by *target* is written to the clipboard. This option only available for X11, OS/2 and Win32 ports of THE.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

PUT, GET

Status:

Complete.

QQUIT − exit from the current file without saving changes

Syntax:

QQuit

Description:

The QQUIT command exits the user from the current file, whether changes made to the file have been saved or not.
The previous file in the ring then becomes the current file.
If the current file is the only file in the ring , THE terminates.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

QUIT

Status:

Complete

QUERY − display various option settings

Syntax:

Query *item*

Description:

The QUERY command displays the various settings for options set by THE.
For a complete list of *item* s that can be extracted, see the section; QUERY, EXTRACT and STATUS .
Results of the QUERY command are displayed at the top of the display window, and ignore the setting of SET MSGLINE .

Compatibility:

XEDIT: Compatible functionality, but not all options.
KEDIT: Compatible functionality, but not all options.

See Also:

STATUS, MODIFY

Status:

Complete.

QUIT − exit from the current file if no changes made

Syntax:

QUIT

Description:

The QUIT command exits the user from the current file, provided that any changes made to the file

have been saved, otherwise an error message is displayed.
The previous file in the ring then becomes the current file.
If the current file is the only file in the ring , THE terminates.

Compatibility:

XEDIT: Does not support return code option.
KEDIT: Compatible.

See Also:

QQUIT

Status:

Complete

READV − read keystrokes and pass to macro

Syntax:

READV *Cmdline [initial text]*
READV *KEY*

Description:

The READV command allows a Rexx macro to interact with the user by accepting either individual keystrokes (*KEY*) or a complete line of text (*Cmdline*).
The READV *Cmdline* can take optional *initial text* to be displayed on the command line.
The 'macro' obtains the entered information by setting Rexx variables. These are set as follows.
KEY option:
readv.0 = 4
readv.1 = name of key (empty if unknown)
readv.2 = ASCII value of key (null if not an ASCII code)
readv.3 = curses key value (or ASCII code if an ASCII code)
readv.4 = shift status (see below)
CMDLINE option:
readv.0 = 1
readv.1 = contents of command line
While editting the command in READV *Cmdline* , any key redefinitions you have made will be in effect. Therefore you can use your "normal" editting keys to edit the line. THE will allow the following commands to be executed while in READV *Cmdline* :
CURSOR LEFT, CURSOR RIGHT, CURSOR DOWN, CURSOR UP,
SOS FIRSTCHAR , SOS ENDCHAR , SOS STARTENDCHAR ,
SOS DELEND , SOS DELCHAR , SOS DELCHAR ,
SOS TABB , SOS TABF , SOS TABWORDB , SOS TABWORDF ,
SOS UNDO , SOS DELWORD , SET INSERTMODE , TEXT
Either of the keys, ENTER, RETURN and NUMENTER will terminate READV *Cmdline* , irrespective of what THE commands have been assigned.
The shift status of the key is an eight character string of 0 or 1; each position represented by the following.
Position 1 1 if INSERTMODE is ON 2 always 0 3 always 0 4 always 0 5 1 if ALT key pressed 6 1 if CTRL key pressed 7 1 if SHIFT key pressed 8 same as position 7

Compatibility:

XEDIT: Similar to READ CMDLINE option.
KEDIT: Compatible.

Status:

Complete.

RECOVER − recover changed or deleted lines

Syntax:

RECover *[n|*]*

Description:

The RECOVER command restores the last *n* , or all * changed or deleted lines back into the body of the file.

Compatibility:

XEDIT: Also recovers changes to lines, not just lines deleted.
KEDIT: Compatible.

Status:

Complete.

REDIT − re−edit the current file

Syntax:

REDIT

Description:

The REDIT command removes the current file from ring, discarding any changes since the file was last saved either explicitly with the SAVE command or implicitly by SET AUTOSAVE , and loads the file back into the ring.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

CANCEL, SAVE

Status:

Complete.

REDRAW − redraw the current screen

Syntax:

REDRAW

Description:

The REDRAW command redraws the current contents of the screen. This is usually used when some outside influence has affected the display.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

REFRESH

Status:

Complete.

REFRESH − refresh the contents of the current screen

Syntax:

REFRESH

Description:

The REFRESH command refreshes what is being displayed on the screen. This is usually used from within a macro to indicate the progress of the macro .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

REDRAW

Status:

Complete.

REPEAT − repeat the last command

Syntax:

REPEat *[target]*

Description:

The REPEAT command advances the current line and executes the last command. It is equivalent to NEXT 1 (or UP 1) and = for the specified number of times specified by *target* .

To determine how many lines on which to execute the last command, THE uses the target to determine how many lines from the current position to the target. This is the number of times the last command is executed.

If the last command to be executed, changes the current line, (because it has a target specification), the next execution of the last command will begin from where the previous execution of last command ended.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Status:

Complete

REPLACE − replace the current line with supplied text

Syntax:

Replace *[text]*

Description:

The REPLACE command replaces the focus line with the supplied *text* .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Status:

Complete.

RESET − cancel the marked block or prefix commands or both

Syntax:

RESet *ALL|Block|Prefix|THIGHlight*

Description:

The RESET command unmarks any marked block or outstanding prefix commands, the text highlight area or all or the above.

Compatibility:

XEDIT: Adds Block, All and THIGHlight options.

KEDIT: Missing some features.

Status:

Complete.

RESTORE − restore various editor settings

Syntax:

RESTore

Description:

The RESTORE command restores various editing settings at the time the command is issued. These settings must have been saved with the PRESERVE command.

If an attempt is made to restore settings that have not been preserved, an error results.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

PRESERVE

Status:

Complete.

REXX − execute Rexx instructions

Syntax:

REXX *rexx instructions*

Description:

The REXX command allows the user to enter Rexx instructions from the command line.

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

MACRO

Status:

Complete.

RGTLEFT − scroll the screen to the left or right

Syntax:

RGTLEFT *[n]*

Description:

The RGTLEFT command scrolls the screen *n* columns to the right if the value of vershift is less than or equal to 0, or if the value of vershift is greater than 0, the screen is scrolled *n* columns to the left.

If *n* is not specified, the screen scrolls by three quarters the number of columns displayed.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

LEFT, RIGHT

Status:

Complete.

RIGHT − scroll the screen to the right

Syntax:

RIght *[n|HALF|FULL]*

Description:

The RIGHT command scrolls the screen to the right.

If *n* is supplied, the screen scrolls by that many columns.

If *HALF* is specified the screen is scrolled by half the number of columns in the filearea .

If *FULL* is specified the screen is scrolled by the number of columns in the filearea .

If no parameter is supplied, the screen is scrolled by one column.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

LEFT, RGTLEFT

Status:

Complete.

SAVE − save changes to current file

Syntax:

SAVE *[filename]*

Description:

The SAVE command writes the current file to disk. If a *filename* is supplied, the current file is saved in that file, unless the file already exists which will result in an error message being displayed. Both 'Alterations' counters on the idline are reset to zero.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SSAVE, FILE, FFILE

Status:

Complete

SCHANGE − selectively change strings

Syntax:

SCHange */string1/string2/ [target] [n] [m]*

Description:

The SCHANGE command changes one string of text to another only after confirming each individual change with the user.

The first parameter to the change command is the old and new string values, seperated by delimiters. The allowable delimiters are '/' '\' and '@' .

The second parameter is the target ; how many lines are to be searched for occurrences of *string1* to be changed.

n determines how many occurrences of *string1* are to be changed to *string2* on each line.

m determines from which occurrence of *string1* on the line changes are to commence.

Compatibility:

XEDIT: Functionally compatible, but syntax different.

KEDIT: Compatible.

Default:

1 1 1

See Also:

CHANGE

Status:

Complete.

SET − execute various set commands

Syntax:

SET *set_command [set_command parameter(s) ...]*

Description:

The SET command is a front end to existing SET commands. It treats the first parameter it receives as a command and executes it.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

SHIFT − move text left or right

Syntax:

SHift *Left|Right [n] [target]*

Description:

The SHIFT command moves text in the direction specified the number of columns *n* for the specified *target* lines.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

SHOWKEY − display current key value and command assignation

Syntax:

SHOWkey *[ALL]*

Description:

With no parameter, the SHOWKEY command prompts the user to enter a key and responds with the key name and associated command (if applicable). To exit from SHOWKEY, press the space bar.
With *ALL* specified, a new file is added to the ring with all default key mappings and any key mappings assigned with the DEFINE command shown. The key mappings are displayed as DEFINE commands.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Status:

Complete.

SORT − sort selected lines in a file

Syntax:

SORT *target [[sort field 1] [...] [sort field 10]]*

Description:

The SORT command sort a portion of a file based on the *sort field* specifications.
A *sort field* specification consists of:
order flag − [Ascending|Descending]
left column − left column of field to sort on
right column − right column of field to sort on
The right column MUST be >= left column.
Only 10 sort fields are allowed.

target can be any valid target including ALL, *, −*, and BLOCK.

Compatibility:

XEDIT: XEDIT only allows ordering flag for all fields

KEDIT: Compatible.

Status:

Complete.

SOS − execute various sos commands

Syntax:

SOS *sos_command [sos_command ...]*

Description:

The SOS command is a front end to existing SOS commands. It treats each parameter it receives as a command and executes it.

The SOS command will execute each command until the list of commands has been exhausted, or until one of the commands returns a non−zero return code.

Compatibility:

XEDIT: XEDIT only permits 1 command

KEDIT: Compatible.

Status:

Complete.

SPLIT − split a line into two lines

Syntax:

SPlit *[ALigned] [Column|CURSOR]*

Description:

The SPLIT command splits the focus line into two lines.

If *Aligned* is specified, the first non−blank character of the new line is positioned under the first non−blank character of the focus line .

If *Aligned* is not specified, the text of the new line starts in column 1.

If *Column* (the default) is specified, the current line is split at the current column location.

If *CURSOR* is specified, the focus line is split at the cursor position.

Compatibility:

XEDIT: Compatible.

Does not support Before/After/Colno options

KEDIT: Compatible.

See Also:

JOIN, SPLTJOIN

Status:

Complete.

SPLTJOIN − split/join two lines

Syntax:

spltjoin

Description:

The SPLTJOIN command splits the focus line into two or joins the focus line with the next line depending on the position of the cursor.

If the cursor is after the last column of a line, the JOIN command is executed, otherwise the SPLIT command is executed.

The text in the new line is aligned with the text in the focus line .

This command can only be used by assigning it to a function key.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

JOIN, SPLIT

Status:

Complete.

SSAVE − force SAVE to specified file

Syntax:

SSave *[filename]*

Description:

The SSAVE command writes the current file to disk. If a *filename* is supplied, the current file is saved in that file, otherwise the current name of the file is used.

If a *filename* is supplied and that *filename* already exists, the previous contents of that *filename* will be replaced with the current file.

Both 'Alterations' counters on the idline are reset to zero.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SAVE, FILE, FFILE

Status:

Complete

STATUS − display current settings of various variables

Syntax:

STATus *[filename]*

Description:

The STATUS command, without the optional *filename* , displays a full screen of current settings for various variables.

With the *filename* , the STATUS command creates a file containing a series of SET commands with the current values of these settings.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible. KEDIT does not support [*filename*] option.

See Also:

QUERY, EXTRACT, MODIFY

Status:

Complete

SUSPEND − suspend THE and return to operating system

Syntax:

SUSPend

Description:

The SUSPEND command suspends the current editing session and returns control to the operating system. Under DOS and OS/2 this is the equivalent of OSNOWAIT . Under UNIX, the process gets placed in the background until it is brought to the foreground.

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

OSNOWAIT

Status:

Complete

TABFILE − edit the file under the file tab or shift FILETABS view

Syntax:

TABFILE *[+|−]*

Description:

The TABFILE makes the file pointed to by the mouse in the FILETABS window the current file.

If run from the command line, without a paramter, the leftmost file displayed in the FILETABS window will be made the current file.

TABFILE −, shifts the files in the FILETABS window one file to the right. TABFILE +, shifts the files in the FILETABS window one file to the left.

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

SET FILETABS

Status:

Complete.

TABPRE − switch between FILEAREA and PREFIX area

Syntax:

tabpre

Description:

The TABPRE command switches the focus of the editor from the filearea to the prefix area and vice versa, depending on which window is currently active.

This command can only be used by assigning it to a function key.

This command will be removed in a future version.

Compatibility:

XEDIT: N/A

KEDIT: Equivalent of SOS LEFTEDGE and SOS PREFIX

See Also:

SOS LEFTEDGE, SOS PREFIX

Status:

Complete.

TAG − displays lines matching target in different colour

Syntax:

TAG *[More|Less] [rtarget]*

Description:

The TAG command is similar to the ALL command, in that it allows lines that match the specified target to be displayed. Where it differs to ALL is that the lines that don 't match are still displayed, but the lines that do match are displayed in the colour specified by SET COLOUR HIGHLIGHT. This target consists of any number of individual targets seperated by ' &' (logical and) or ' | ' (logical or). '

For example, to display all lines in a file that contain the strings 'ball' and 'cat' on the same line or the named lines .fred or .bill, use the following command

TAG /ball/ &/cat/ | .fred | .bill

Logical operators act left to right, with no precedence for .

TAG without any arguments displays all lines without any highlighting.

If SET HIGHLIGHT is not set to TAGGED, then if the specified *rtarget* is found, SET HIGHLIGHT is set to TAGGED.

Compatibility:

XEDIT: N/A

KEDIT: Compatible. MORE and LESS options not implemented.

See Also:

ALL, SET HIGHLIGHT, SET COLOUR

Status:

Complete.

TEXT − simulate keyboard entry of characters

Syntax:

TEXT *text*

Description:

The TEXT command simulates the entry of *text* from the keyboard. This command is actually called when you enter text from the keyboard.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Does not allow trailing spaces in text.

Status:

Complete.

THE − edit another file or switch to next file

Syntax:

THE *[filename]*

Description:

The THE command allows the user to edit another 'file' . The new file is placed in the file ring . The previous file being edited remains in memory and can be returned to by issuing a THE command without any parameters. Several files can be edited at once, and all files are arranged in a ring, with subsequent THE commands moving through the ring, one file at a time.

Compatibility:

XEDIT: Does not provide options switches.

KEDIT: Does not provide options switches.

See Also:

XEDIT, EDIT, KEDIT

Status:

Complete.

TOASCII − convert the target from EBCDIC to ASCII

Syntax:

TOASCII *[target]*

Description:

The TOASCII command converts the characters in the target from EBCDIC encoding to ASCII coding.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Status:

Complete.

TOP − move to the top of the file

Syntax:

TOP

Description:

The TOP command moves to the very start of the current file. The Top−of−File line is set to the current line .

TOP is equivalent to BACKWARD *.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

BACKWARD, BOTTOM

Status:

Complete

UP − move backward in the file a number of lines

Syntax:

Up *[relative target]*

Description:

The UP command moves the current line backwards the number of lines specified by the relative target . This relative target can only be a positive integer or the character "*".

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1

See Also:

NEXT, DOWN

Status:

Complete.

UPPERCASE − change lowercase characters to uppercase

Syntax:

UPPercase *[target]*

Description:

The UPPERCASE command changes all lowercase characters in all lines up to the *target* line to uppercase. All other characters remain untouched.

Compatibility:

XEDIT: Equivalent of UPPERCAS

KEDIT: Compatible.

See Also:

LOWERCASE

Status:

Complete.

XEDIT − edit another file or switch to next file

Syntax:

Xedit *[file]*

Description:

The XEDIT command allows the user to edit another *file* . The new file is placed in the file ring . The previous file being edited remains in memory and can be returned to by issuing an XEDIT command without any parameters. Several files can be edited at once, and all files are arranged in a ring, with subsequent XEDIT commands moving through the ring, one file at a time.

Compatibility:

XEDIT: Does not provide options switches.

KEDIT: Does not provide options switches.

See Also:

EDIT, THE, KEDIT

Status:

Complete.

? − retrieve − return the next/prior command on the command line

Syntax:

?[+|?...]

Description:

The ? command returns the next or prior command from the command line ring and displays it on the command line.

With the [+] argument, the next command in the command ring is retrieved.

With no arguments, the previous command entered on the command line is retrieved.

With multiple, concatenated ?s as argument, the pervious command entered on the command line is retrieved corresponding to the number of ?s entered.

For Example: The command; ????? will retrieve the fifth last command entered.

Compatibility:

XEDIT: Compatible. Adds extra support for multiple ?s.

KEDIT: See below..

This command is bound to the up and down arrows when on the command line depending on the setting of SET CMDARROWS .

See Also:

SET CMDARROWS

Status:

Complete.

= − re−execute the last command issued on the command line

Syntax:

=

Description:

The = command retrieves the most recently issued command from the command line and re−executes it.

Compatibility:

XEDIT: Does not support optional [subcommand] option.

KEDIT: Does not support optional [command] option.

Status:

Complete.

! − execute an operating system command

Syntax:

! *[command]*

Description:

The ! command executes the supplied operating system *command* or runs an interactive shell if no *command* is supplied.

Compatibility:

XEDIT: N/A
KEDIT: Equivalent to DOS command.

See Also:

DOS, OS

Status:

Complete.

&− execute and re−display command

Syntax:

Description:

The &command executes the supplied *command* in the normal way, but when the command completes, instead of clearing the THE command line, the command, and the &are re−displayed. This makes it easy to repeat the same command, or make changes to it.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Status:

Complete.

Return to Table of Contents

SET COMMAND REFERENCE

SET ALT − change alteration counts

Syntax:

[SET] ALT *[n] [m]*

Description:

The SET ALT command allows the user to change the alteration counts. This command is usually called from within a macro.

The first number; *n* sets the number of changes since the last AUTOSAVE was issued.

The second number; *m* sets the number of changes since the last SAVE or SSAVE command was issued.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

OFF

See Also:

SET EQUIVCHAR

Status:

Complete.

SET ARBCHAR − set arbitrary character(s) for targets

Syntax:

[SET] ARBchar *ON|OFF [char1] [char2]*

Description:

Set the character to use as an 'arbitrary character' in string targets. The first arbitrary character matches a group of zero or more characters, the second will match exactly one character.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: Compatible.

Single arbitrary character not supported.

KEDIT: Compatible.

Default:

Off $?

See Also:

SET EQUIVCHAR

Status:

Complete.

SET AUTOCOLOR − specifies which parser to use for syntax highlighting

Syntax:

[SET] AUTOCOLOR *mask parser [MAGIC]*

Description:

The SET AUTOCOLOR command allows the user to specify which syntax highlighting parser is to be used for which file masks.

The *parser* argument specifies a syntax highlighting parser that already exists, either as a default parser , or added by the user with SET PARSER . The special parser name of '*NULL' can be specified; this will effectively remove the association between the parser and the file mask.

The *mask* argument specifies the file mask (or magic number) to associate with the specified parser. The *mask* can be any valid file mask for the operating system. eg *.c fred.* joe.?

If the *magic* option is specified, the *mask* argument refers to the last element of the magic number that is specified in the first line of a Unix shell script comment. eg if the first line of a shell script contains: #!/usr/local/bin/rexx then the file mask argument would be specified as "rexx".

Compatibility:

XEDIT: N/A

KEDIT: Similar. KEDIT does not have MAGIC option.

Default:

See QUERY AUTOCOLOR

See Also:

SET COLORING, SET ECOLOUR, SET PARSER

Status:

Complete.

SET AUTOCOLOUR − specifies which parser to use for syntax highlighting

Syntax:

[SET] AUTOCOLOR *mask parser [MAGIC]*

Description:

The SET AUTOCOLOUR command is a synonym for the SET AUTOCOLOR command.

Compatibility:

XEDIT: N/A

KEDIT: Similar. KEDIT does not have MAGIC option.

Default:

See QUERY AUTOCOLOR

See Also:

SET AUTOCOLOR

Status:

Complete.

SET AUTOSAVE − set autosave period

Syntax:

[SET] AUtosave *n|OFF*

Description:

The SET AUTOSAVE command sets the interval between automatic saves of the file, or turns it off altogether. The interval *n* refers to the number of alterations made to the file. Hence a value of 10 for *n* would result in the file being automatically saved after each 10 alterations have been made to the file.

It is not possible to set AUTOSAVE for 'psuedo' files such as the directory listing 'file' , Rexx output 'file' and the key definitions 'file'

Compatibility:

XEDIT: Does not support [mode] option.

KEDIT: Compatible.

Default:

OFF

Status:

Complete.

SET AUTOSCROLL − set rate of automatic horizontal scrolling

Syntax:

[SET] AUTOSCroll *n|OFF|Half*

Description:

The SET AUTOSCROLL allows the user to set the rate at which automatic horizontal scrolling occurs.

When the cursor reaches the last (or first) column of the filearea the filearea can automatically scroll if AUTOSCROLL is not *OFF* and a CURSOR RIGHT or CURSOR LEFT command is issued. How many columns are scrolled is determined by the setting of AUTOSCROLL.

If AUTOSCROLL is set to *HALF* , then half the number of columns in the filearea window are scrolled. Any other value will result in that many columns scrolled, or the full width of the filearea window if the set number of columns is larger.

Autoscrolling does not occur if the key pressed is assigned to CURSOR SCREEN LEFT or RIGHT, which is the case if SET COMPAT XEDIT key defintions are active.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

HALF

Status:

Complete.

SET BACKUP − indicate if a backup copy of the file is to be kept

Syntax:

[SET] BACKup *OFF|TEMP|KEEP|ON|INPLACE*

Description:

The SET BACKUP command allows the user to determine if a backup copy of the original file is to be kept when the file being edited is saved or filed.

KEEP and *ON* options are the same. *ON* is kept for compatability with previous versions of THE.

With *OFF* , the file being written to disk will replace an existing file. There is a chance that you will end up with neither the old version of the file or the new one if problems occur while the file is being written.

With *TEMP* or *KEEP* options, the file being written is first renamed to the filename with a .bak extension. The file in memory is then written to disk. If *TEMP* is in effect, the backup file is then deleted.

With *INPLACE* , the file being written is first copied to a file with a .bak extension. The file in memory is then written to disk in place of the original. This option ensures that all operating system file attributes are retained.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

KEEP

See Also:

FILE, FFILE, SAVE, SSAVE

Status:

Complete.

SET BEEP − turn on or off the audible alarm when displaying errors

Syntax:

[SET] BEEP *ON|OFF*

Description:

The SET BEEP command allows the user to determine if an audible alarm is sounded when an error is displayed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

OFF

Status:

Complete.

SET BOUNDMARK − set bounds marker display

Syntax:

[SET] BOUNDMARK *OFF|Zone|TRunc|MARgins|TABs|Verify*

Description:

The BOUNDMARK command indicates if boundary markers are to be displayed and if so, where. Boundary markers are vertical lines drawn before or after certain columns within the filearea . This command only has a visible effect on GUI platforms, currently only the X11 port.

OFF turns off the display of boundary markers.

ZONE turns on the display of boundary markers, before the zone start column and after the zone end column.

TRUNC turns on the display of boundary markers, after the truncation column. Not supported.

MARGINS turns on the display of boundary markers, before the left margin and after the right margin.

TABS turns on the display of boundary markers, before each tab column.

VERIFY turns on the display of boundary markers, before each verify column. Not supported.

Compatibility:

XEDIT: N/A
KEDIT: Compatible, but no support for TRUNC or VERIFY option.

Default:

Zone

Status:

Incomplete

SET CASE − set case sensitivity parameters

Syntax:

[SET] CASE *Mixed|Lower|Upper [Respect|Ignore] [Respect|Ignore] [Respect|Ignore]*

Description:

The CASE command sets the editor 's handling of the case of text.'

The first option (which is mandatory) controls how text is entered by the user. When *LOWER* or *UPPER* are in effect, the shift or caps lock keys have no effect on the text being entered. When *MIXED* is in effect, text is entered in the case set by the use of the shift and caps lock keys.

The second option determines how the editor determines if a string target matches text in the file when the target is used in a LOCATE command. With *IGNORE* in effect, a match is found irrespective of the case of the target or the found text. The following strings are treated as equivalent: the THE The ThE... With *RESPECT* in effect, the target and text must be the same case. Therefore a target of 'The' only matches text containing 'The' , not 'THE' or 'ThE' etc.

The third option determines how the editor determines if a string target matches text in the file when the target is used in a CHANGE command. With *IGNORE* in effect, a match is found irrespective of the case of the target or the found text. The following strings are treated as equivalent: the THE The ThE... With *RESPECT* in effect, the target and text must be the same case. Therefore a target of 'The' only matches text containing 'The' , not 'THE' or 'ThE' etc.

The fourth option determines how the editor determines the sort order of upper and lower case with the SORT command. With *IGNORE* in effect, upper and lower case letters are treated as equivalent. With *RESPECT* in effect, upper and lower case letters are treated as different values and uppercase characters will sort before lowercase characters.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: Adds support for case significance in CHANGE commands.

KEDIT: Adds support for LOWER option.

Both: Adds support for case significance in SORT command.

Default:

Mixed Ignore Respect Respect

See Also:

SET EQUIVCHAR

Status:

Complete

SET CLEARERRORKEY − specify which key clears the message line

Syntax:

[SET] CLEARErrorkey *|*keyname*

Description:

The SET CLEARERRORKEY command allows the user to specify which key clears the message line. By default, any key pressed will cause the message line to be cleared. The keyname specified is the name returned via the SHOWKEY command.

As the QUERY command also uses the same mechanism for displaying its results as errors, then this command affects when results from the QUERY command are cleared.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

*

Status:

Complete

SET CLEARSCREEN − indicate if the screen is to be cleared on exit

Syntax:

[SET] CLEARScreen *ON*|*OFF*

Description:

The SET CLEARSCREEN command allows the user to request that the screen be cleared on exit from THE.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

OFF

Status:

Complete

SET CLOCK − turn on or off display of time on status line

Syntax:

[SET] CLOCK *ON|OFF*

Description:

The SET CLOCK command turns on or off the display of the time on the status line .

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

ON

Status:

Complete

SET CMDARROWS − sets the behaviour of the up and down arrow keys

Syntax:

[SET] CMDArrows *Retrieve|Tab*

Description:

The SET CMDARROWS command determines the action that occurs when the up and down arrows keys are hit while on the command line .

RETRIEVE will set the up and down arrows to retrieve the last or next command entered on the command line .

TAB will set the up and down arrows to move to the last or first line respectively of the main window.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

RETRIEVE

See Also:

CURSOR, ?

Status:

Complete.

SET CMDLINE − sets the position of the command line.

Syntax:

[SET] CMDline *ON|OFF|Top|Bottom*

Description:

The SET CMDLINE command sets the position of the command line , either at the top of the screen, the bottom of the screen or off.

Compatibility:

XEDIT: Compatible.

CMDLINE ON is equivalent to CMDLINE Bottom

KEDIT: Compatible.

Default:

BOTTOM

Status:

Complete.

SET COLOR − set colours for display

Syntax:

[SET] COLOR *area [modifier[...]] [foreground] [ON] [background]*

[SET] COLOR *area [modifier[...]] ON|OFF*

Description:

The SET COLOR command changes the colours or display attributes of various display areas in THE.

Valid values for *area* :

ALERT − alert boxes; see ALERT

Arrow − command line prompt

Block − marked block

BOUNDmarker− bound markers (GUI platforms only)

CBlock − current line if in marked block

CHIghlight − highlighted line if the same as current line

Cmdline − command line

CTofeof − as for TOfeof if the same as current line

CUrline − the current line

DIALOG − dialog boxes; see DIALOG

Divider − dividing line between vertical split screens

Filearea − area containing file lines

GAP − the gap between the prefix area and filearea

CGAP − the gap between the prefix area and filearea − current

HIghlight − highlighted line

Idline − line containing file specific info

Msgline − error messages

Nondisp − Non−display characters (SET ETMODE OFF)

Pending − pending commands in prefix area

PRefix − prefix area

CPRefix − prefix area if the same as current line

Reserved − default for reserved line

Scale − line showing scale line

SHadow − hidden line marker lines

SLK − soft label keys

STatarea − line showing status of editing session

Tabline − line showing tab positions

TOfeof − Top−of−File line and Bottom−of−File line

* − All area (second format only)

Valid values for *foreground* and *background* :

BLAck

BLUe

Brown

Green

GRAy

GREy

Cyan

RED

Magenta

Pink

Turquoise

Yellow

White

Valid values for *modifier* :

NORmal

BLInk

BOld

BRIght
High
REVerse
Underline
DARK
Italic − only available on X11 port with valid Italic font

The second format of this command allows the user to turn on or off any of the valid modifiers.

It is an error to attempt to set a colour on a mono display.

Compatibility:

XEDIT: Functionally compatible. See below.
KEDIT: Functionally compatible. See below.
Does not implement all modifiers.

Default:

Depends on compatibility mode setting and monitor type.

See Also:

SET COMPAT, SET COLOUR, SET ECOLOUR

Status:

Complete.

SET COLOUR − set colours for display

Syntax:

[SET] COLOUR *area [modifier[...]] [foreground] [on background]*
[SET] COLOUR *area [modifier[...]] ON|OFF*

Description:

The SET COLOUR command is a synonym for the SET COLOR command.

Compatibility:

XEDIT: Functionally compatible. See below.
KEDIT: Functionally compatible. See below.
Does not implement all modifiers.

Default:

Depends on compatibility mode setting and monitor type.

See Also:

SET COLOR

Status:

Complete.

SET COLORING − enable or disable syntax highlighting

Syntax:

[SET] COLORING *ON|OFF [AUTO|parser]*

Description:

The SET COLORING command allows the user to turn on or off syntax highlighting for current file. It also allows the parser used to be specified explicitly, or automatically determined by the file extension or magic number .

ON turns on syntax highlighting for the current file, OFF turns it off.

AUTO determines the parser to use for the current file based on the file extension. The parser to use is controlled by the SET AUTOCOLOR command.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON AUTO

See Also:

SET COLOURING, SET ECOLOUR, SET AUTOCOLOR, SET PARSER

Status:

Complete.

SET COLOURING − enable or disable syntax highlighting

Syntax:

[SET] COLOURING *ON|OFF [AUTO|parser]*

Description:

The SET COLOURING command is a synonym for the SET COLORING command.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON AUTO

See Also:

SET COLORING

Status:

Complete.

SET COMPAT − set compatibility mode

Syntax:

[SET] COMPat *The|Xedit|Kedit|KEDITW|Ispf|= [The|Xedit|Kedit|KEDITW|Ispf|=] [The|Xedit|Kedit|KEDITW|Ispf|=]*

Description:

The SET COMPAT command changes some settings of THE to make it more compatible with the look and/or feel of XEDIT, KEDIT, KEDIT for Windows, or ISPF.

This command is most useful as the first SET command in a profile file. It will change the default settings of THE to initially look like the chosen editor. You can then make any additional changes in THE by issuing other SET commands.

It is recommended that this command NOT be executed from the command line, particularly if you have 2 files being displayed at the same time. Although the command works, things may look and behave strangely :−)

The first parameter affects the look of THE, the second parameter affects the feel of THE, and the third parameter determines which default function key settings you require.

Any of the parameters can be specified as =, which will not change that aspect of THE 's compatability.'

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

THE THE THE

Status:

Complete.

SET CTLCHAR − define control character attributes

Syntax:

[SET] CTLchar *OFF*

[SET] CTLchar *char Escape | OFF*

[SET] CTLchar *char Protect|Noprotect [modifier[...]] [fore [ON back]]*

Description:

The SET CTLCHAR command defines control characters to be used when displaying a reserved line . Control characters determine how parts of a reserved line are displayed.

See SET COLOUR for valid values for *modifier* , *fore* and *back* .

The *Protect* and *Noprotect* arguments are ignored.

Compatibility:

XEDIT: Similar, but does not support all parameters.

KEDIT: N/A.

Default:

OFF

See Also:

SET COLOUR, SET RESERVED

Status:

Complete.

SET CURLINE − set position of current line on screen

Syntax:

[SET] CURLine *M[+n|−n] | [+|−]n*

Description:

The SET CURLINE command sets the position of the current line to the physical screen line specified by supplied arguments.

The first form of parameters is:

M[+n|−n]

this sets the current line to be relative to the middle of
the screen. A positive value adds to the middle line number,
a negative subtracts from it.
eg. M+3 on a 24 line screen will be line 15
M−5 on a 24 line screen will be line 7

The second form of parameters is:

[+|−]n

this sets the current line to be relative to the top of the
screen (if positive or no sign) or relative to the bottom
of the screen if negative.
eg. +3 or 3 will set current line to line 3
−3 on a 24 line screen will be line 21

If the resulting line is outside the bounds of the screen the position of the current line will become the middle line on the screen.

It is an error to try to position the CURLINE on the same line as a line already allocated by one of SET HEXSHOW , SET RESERVED , SET SCALE or SET TABLINE .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

M

Status:

Complete.

SET CURSORSTAY − set on or off the behaviour of the cursor on a scroll

Syntax:

[SET] CURSORSTay *ON|OFF*

Description:

The SETCURSORSTAY command allows the user to set the behaviour of the cursor when the file is scrolled with a FORWARD or BACKWARD command.

Before this command was introduced, the position of the cursor after the file was scrolled depended on SET COMPAT ; for THE, the cursor moved to the current line, for XEDIT and KEDIT modes the cursor stayed on the same screen line.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

ON

Status:

Complete.

SET DEFSORT − specify the order in which files appear in DIR.DIR

Syntax:

[SET] DEFSORT *OFF|DIRectory|Size|Date|Time|Name [Ascending|Descending]*

Description:

The SET DEFSORT command allows the user to determine the order in which files appear in a DIR.DIR file.

Directory specifies that directories within the current directory are shown before other files.

Size specifies that the size of the file determines the order in which files are displayed.

Date specifies that the date of the last change to the file determines the order in which files are displayed. If the dates are the same, the time the file was last changed is used as a secondary sort key.

Time specifies that the time of the file determines the order in which files are displayed.

Name specifies that the name of the file determines the order in which files are displayed. This is the default. Files are sorted by name as a secondary sort key when any of the above options are specified and two files have equal values for that sort option.

OFF indicates that no ordering of the files in the directory is performed. On directories with a large number of files, this option results in a displayed DIR.DIR file much quicker than any sorted display.

The second parameter specifies if the sort order is ascending or descending.

This command does not affect how any current DIR.DIR file is shown but is applicable the next time a directory is displayed as a result of a DIR or LS command.

Compatibility:

XEDIT: N/A
KEDIT: Similar in functionality.

Default:

NAME ASCENDING

Status:

Complete.

SET DIRINCLUDE − set the file mask for directory command

Syntax:

[SET] DIRInclude *

[SET] DIRInclude *[Normal] [Readonly] [System] [Hidden] [Directory]*

Description:

The DIRINCLUDE command sets the file mask for files that will be displayed on subsequent DIRECTORY commands. The operand "*" will set the mask to all files, the other options will set the mask to include those options specified together with "normal" files eg.

DIRINCLUDE R S

will display readonly and system files together with "normal" files the next time the DIRECTORY command is issued.

The effects of DIRINCLUDE are ignored in the Unix version.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

*

See Also:

DIRECTORY, LS

Status:

Complete.

SET DISPLAY − specify which level of lines to display

Syntax:

[SET] DISPlay *n [m|*]*

Description:

The SET DISPLAY command sets the selection level for lines to be displayed on the screen.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

0 0

See Also:

SET SCOPE, SET SELECT, ALL

Status:

Complete.

SET ECOLOR − set colors for syntax highlighting

Syntax:

[SET] ECOLOR *char [modifier[...]] [foreground] [on background]*
[SET] ECOLOR *char [modifier[...]] ON|OFF*

Description:

The SET ECOLOR command allows the user to specify the colors of each category of items used in syntax highlighting.

char refers to one of the following valid values:

A − comments
B − strings
C − numbers
D − keywords
E − labels
F − preprocessor directives
G − header lines
H − extra right paren, matchable keyword (N/A)
I − level 1 paren
J − level 1 matchable keyword (N/A)
K − level 1 matchable preprocessor keyword (N/A)
L − level 2 paren, matchable keyword (N/A)
M − level 3 paren, matchable keyword (N/A)
N − level 4 paren, matchable keyword (N/A)
O − level 5 paren, matchable keyword (N/A)
P − level 6 paren, matchable keyword (N/A)
Q − level 7 paren, matchable keyword (N/A)
R − level 8 paren or higher, matchable keyword (N/A)
S − incomplete string
T − HTML markup tags
U − HTML character/entity references

V − Builtin functions
W − not used
X − not used
Y − not used
Z − not used
1 − alternate keyword color 1
2 − alternate keyword color 2
3 − alternate keyword color 3
4 − alternate keyword color 4
5 − alternate keyword color 5
6 − alternate keyword color 6
7 − alternate keyword color 7
8 − alternate keyword color 8
9 − alternate keyword color 9

N/A indicates that this capability is not yet implemented.

For valid values for *modifier* , *foreground* and *background* see SET COLOR .

The second format of this command allows the user to turn on or off any of the valid modifiers.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

See QUERY ECOLOR

See Also:

SET COLORING, SET AUTOCOLOR, SET PARSER, SET COLOR

Status:

Complete.

SET ECOLOUR − set colours for syntax highlighting

Syntax:

[SET] ECOLOUR *char [modifier[...]] [foreground] [on background]*
[SET] ECOLOUR *char [modifier[...]] ON|OFF*

Description:

The SET ECOLOUR command allows the user to specify the colours of each category of items used in syntax highlighting.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

See QUERY ECOLOR

See Also:

SET COLOURING, SET AUTOCOLOUR, SET PARSER, SET COLOUR

Status:

Complete.

SET EOLOUT − set end of line terminating character(s)

Syntax:

[SET] EOLout *CRLF|LF|CR|NONE*

Description:

The EOLOUT command allows the user to specify the combination of characters that terminate a line. Lines of text in Unix files are usually terminated with a *LF* , DOS file usually end with a *CR* and *LF* combination. Files on the Apple Macintosh are usually terminated with a *CR* .

The *NONE* option can be used to specify that no end of line character is written.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

LF − UNIX
CRLF − DOS/OS2/WIN32
NONE − if THE started with −u option

Status:

Complete.

SET EQUIVCHAR − set the equivalence character

Syntax:

[SET] EQUIVChar *char*

Description:

The SET EQUIVChar command allows the user to change the character that is used to specify eqivalence in command parameters.

In many THE commands, an equivalence character, usually '=' , can be used as a parameter to default to values in the current file or view.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

=

Status:

Complete.

SET ETMODE − indicate if extended display mode is possible

Syntax:

[SET] ETMODE *ON|OFF [character list]*

Description:

The SET ETMODE command allows the user to specify which characters in a character set are to be displayed as their actual representation.

Those characters not explicitly specified to be displayed as they are represented, will be displayed as the SET NONDISP character in the colour specified by SET COLOUR NONDISP. Characters below 32, will be displayed with an alphabetic character representing the "control" code.

eg. character code with a value of 7, will display as "G" in the colour specified by SET COLOUR NONDISP.

ON with no optional *character list* will display ALL characters as their actual representation.

OFF with no optional *character list* will display control characters below ASCII 32, as a "control" character; characters greater than ASCII 126 will be displayed as the SET NONDISP characters. On ASCII based machines, [SET] ETMODE OFF is equivalent to [SET] ETMODE ON 32−126. On EBCDIC based machines [SET] ETMODE OFF is equivalent to [SET] ETMODE ON ??−??

The *character list* is a list of positive numbers between 0 and 255 (inclusive). The format of this character list can be either a single number; eg. 124, or a range of numbers specified; eg. 32−126. (The first number must be less than or equal to the second number).

As an example; ETMODE ON 32−127 160−250 would result in the characters with a decimal value between 32 and 127 inclusive and 160 and 250 inclusive being displayed as their actual representation (depending on the current font), and the characters between 0 and 31 inclusive, being displayed as an equivalent "control" character; characters between 128 and 159 inculsive and 250 to 255 being displayed with the SET NONDISP character.

Up to 20 character specifiers (single number or range) can be specified.

Compatibility:

XEDIT: Similar function but deals with Double−Byte characters
KEDIT: N/A

Default:

ON − DOS/OS2/WIN32
ON 32−255 − X11
OFF − UNIX/AMIGA/QNX

See Also:

SET NONDISP, SET COLOUR

Status:

Complete.

SET FEXT − change the extension of the existing file

Syntax:

[SET] FExt *ext*
[SET] FType *ext*

Description:

The SET FEXT command allows the user to change the path of the file currently being edited.
The 'path' parameter can be specified with or without the trailing directory seperator. Under DOS, OS/2 and Windows ports, the drive letter is considered part of the file 's path.'
See SET FILENAME for a full explanation of THE 's definitions of fpath, filename, fname, fext and fmode.'
It is not possible to use this command on pseudo files.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SET FNAME, SET FILENAME, SET FTYPE, SET FMODE

Status:

Complete.

SET FILENAME − change the filename of the file being edited

Syntax:

[SET] FILEName *filename*

Description:

The SET FILEName command allows the user to change the filename of the file currently being edited.
In THE, a fully qualified file name consists of a file path and a file name. THE treats all characters up to and including the last directory seperator (usually / or \) as the file 's path. From the first character after the end of the file' s path, to the end of the fully qualified file name is the file name.
A file name is further broken down into a fname and fext. The fname of a file consists of all characters from the start of the filename up to but not including the last period (if there is one). The fext of a file consists of all characters from the end of the filename up to but not including the last period. If there is no period in the filename then the fext is empty.
The fmode of a file is equivalent to the drive letter of the file 's path. This is only valid under DOS, OS/2 and Windows ports.'
Some examples.

```
Full File Name     File            File     Fname  Fext     Fmode
                   Path            Name
-----------------------------------------------------------------
/usr/local/bin/the /usr/local/bin/ the      the             N/A
c:\tools\the.exe   c:\tools\       the.exe  the    exe      c
/etc/a.b.c         /etc/           a.b.c    a.b    c        N/A
```

A limited amount of validation of the resulting file name is carried out by this command, but some errors in the file name will not be evident until the file is saved.

A leading "=" indicates that the fname portion of the current file name is be retained. This is equivalent to the command SET FEXT . A trailing "=" indicates that the fext portion of the current file name is to be retained. This is equivalent to the command SET FNAME .

Only one "=" is allowed in the parameter.

Some examples.

File Name	Parameter	New File Name	
a.b.c	fred.c=	fred.c.c	SET FNAME fred.c
a.b.c	fred.c.=	fred.c..c	SET FNAME fred.c.
a.b.c	=fred	a.c.fred	SET FEXT fred
a.b.c	=.fred	a.c..fred	SET FEXT .fred
a	=d	a.d	SET FEXT d
a.b.c	=	a.b.c	does nothing

It is not possible to use this command on pseudo files.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

SET FPATH, SET FNAME, SET FEXT, SET FMODE, SET EQUIVCHAR

Status:

Complete.

SET FILETABS − determine if and where where file tabs are positioned

Syntax:

[SET] FILETABS *ON*|*OFF*

Description:

The SET FILETABS command allows the user to determine if file tabs are to be displayed and where. FILETABS is a single line at the top of the display showing all files currently in the ring, except the current file. It provides a mechanism where the user running THE with mouse support can simply click on the filename in the FILETABS line to change focus to that file.

The colour of the file tabs can be set with SET COLOUR FILETABS. The colour of the file dividers can be set with SET COLOUR FILETABSDIV.

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

SET COLOUR, TABFILE

Status:

Complete.

SET FMODE − change the drive letter of the existing file

Syntax:

[SET] FMode *d[:]*

Description:

The SET FMode command allows the user to change the drive letter of the file currently being edited.

This command is only valid under the DOS, OS/2 and Windows ports.

See SET FILENAME for a full explanation of THE 's definitions of fpath, filename, fname, fext and fmode.'

It is not possible to use this command on pseudo files.

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SET FNAME, SET FILENAME, SET FEXT, SET FPATH

Status:

Complete.

SET FNAME − change the filename of the file being edited

Syntax:

[SET] FName *filename*

Description:

The SET FNAME command allows the user to change the fname of the file currently being edited.

See SET FILENAME for a full explanation of THE 's definitions of fpath, filename, fname, fext and fmode.'

A limited amount of validation of the resulting file name is carried out by this command, but some errors in the file name will not be evident until the file is saved.

It is not possible to use this command on pseudo files.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SET FPATH, SET FILENAME, SET FEXT, SET FMODE

Status:

Complete.

SET FPATH − change the path of the existing file

Syntax:

[SET] FPath *path*

Description:

The SET FPATH command allows the user to change the path of the file currently being edited.

The *path* parameter can be specified with or without the trailing directory seperator. Under DOS, OS/2 and Windows ports, the drive letter is considered part of the file 's path.'

See SET FILENAME for a full explanation of THE 's definitions of fpath, filename, fname, fext and fmode.'

It is not possible to use this command on pseudo files.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SET FNAME, SET FILENAME, SET FEXT, SET FMODE

Status:

Complete.

SET FTYPE − change the extension of the existing file

Syntax:

[SET] FType *ext*

Description:

The SET FTYPE is a synonym for SET FEXT .

Compatibility:

XEDIT: N/A

KEDIT: N/A

See Also:

SET FNAME, SET FILENAME, SET FEXT, SET FMODE

Status:

Complete.

SET FULLFNAME − specify if complete filename to be displayed

Syntax:

[SET] FULLFName *ON*|*OFF*

Description:

The SET FULLFNAME command allows the user to determine if the fully qualified filename is displayed on the IDLINE or just the FNAME component. See SET FILENAME for a full explanation of THE 's definitions of fpath, filename, fname, fext and fmode.'

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

ON

Status:

Complete.

SET HEADER − turn on or off syntax highlighting headers

Syntax:

[SET] HEADer *section ON*|*OFF*

Description:

The SET HEADER command allows fine tuning of which sections of a TLD file are to be applied for the current view.
section refers to one of the following headers that can be specified in a TLD file: NUMBER, COMMENT, STRING, KEYWORD, FUNCTION, HEADER, LABEL, MATCH, COLUMN, POSTCOMPARE, MARKUP. *section* can also be specified as '*' , in which case, all headers are applied or not applied.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

* ON

See Also:

SET PARSER, SET COLORING, SET AUTOCOLOR

Status:

Complete.

SET HEX − set how hexadecimal strings are treated in string operands

Syntax:

[SET] HEX *ON*|*OFF*

Description:

The SET HEX set command determines whether hexadecimal strings are treated as such in string operands.
With the *ON* option, any string operand of the form /x '31 32 33' / or /d '49 50 51' / will be converted to /123/ before the command is executed.
With the *OFF* option, no conversion is done.

This conversion should work wherever a string operand is used in any command.

Compatibility:

XEDIT: Adds support for decimal representation. See below.
KEDIT: Compatible. See below.
Spaces must seperate each character representation.

Default:

OFF

Status:

Complete.

SET HEXDISPLAY − turn on or off display of character under cursor

Syntax:

[SET] HEXDISPlay *ON|OFF*

Description:

The SET HEXDISPLAY command turns on or off the display of the character under the cursor on the status line .

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON

Status:

Complete

SET HEXSHOW − turn on or off hex display of current line

Syntax:

[SET] HEXShow *ON|OFF [M[+n|−n]|[+|−]n]*

Description:

The SET HEXShow command indicates if and where a hexadecimal representation of the current line will be displayed.
The first form of parameters is:
M[+n|−n]
this sets the hexshow line to be relative to the middle of
the screen. A positive value adds to the middle line number,
a negative subtracts from it.
eg. M+3 on a 24 line screen will be line 15
M−5 on a 24 line screen will be line 7
The second form of parameters is:
[+|−]n
this sets the hexshow line to be relative to the top of the
screen (if positive or no sign) or relative to the bottom
of the screen if negative.
eg. +3 or 3 will set current line to line 3
−3 on a 24 line screen will be line 21
If the resulting line is outside the bounds of the screen the position of the hexshow line will become the middle line on the screen.
The position argument specifies the position of the first line of the hexadecimal display.
It is an error to try to position the HEXSHOW lines on the same line as SET CURLINE .

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

OFF 7

Status:

Complete

SET HIGHLIGHT − specify which lines (if any) are to be highlighted

Syntax:

[SET] HIGHlight *OFF|TAGged|ALTered|SELect n [m]*

Description:

The SET HIGHLIGHT command allows for the user to specify which lines are to be displayed in the highlighted colour.

OFF turns all highlighting display off

TAGGED displays all tagged lines in the highlight colour.

ALTERED displays all lines that have been added or changed in the current session in the highlight colour.

'SELECT n [m]' displays all lines with the specified selection level in highlight colour.

Compatibility:

XEDIT: N/A

KEDIT: Compatible

Default:

OFF

See Also:

SET SELECT, TAG, SET LINEFLAG

Status:

Ccomplete.

SET IDLINE − specify if IDLINE is displayed

Syntax:

[SET] IDline *ON|OFF*

Description:

The SET IDLINE set command determines if the idline for a file is displayed or not.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

ON

Status:

Complete

SET IMPCMSCP − set implied operating system command processing

Syntax:

[SET] IMPcmscp *ON|OFF*

Description:

The SET IMPCMSCP command is used to toggle implied operating system command processing from the command line. By turning this feature on you can then issue an operating system command without the need to prefix the operating system command with the OS command.

Compatibility:

XEDIT: Compatible.

KEDIT: N/A

Default:

ON

See Also:

SET IMPOS

Status:

Complete.

SET IMPMACRO − set implied macro command processing

Syntax:

[SET] IMPMACro *ON|OFF*

Description:

The SET IMPMACRO command is used to toggle implied macro processing from the command line. By turning this feature on you can then issue a macro command without the need to prefix the macro name with the MACRO command.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON

See Also:

MACRO, SET MACROPATH

Status:

Complete.

SET IMPOS − set implied operating system command processing

Syntax:

[SET] IMPOS *ON|OFF*

Description:

The SET IMPOS command is used to toggle implied operating system command processing from the command line. By turning this feature on you can then issue an operating system command without the need to prefix the operating system command with the OS command.

Compatibility:

XEDIT: Compatible.
KEDIT: N/A

Default:

ON

See Also:

SET IMPCMSCP

Status:

Complete.

SET INPUTMODE − set input mode behaviour

Syntax:

[SET] INPUTMode *OFF|FUll|LIne*

Description:

The SET INPUTMODE command changes the way THE handles input.
When INPUTMODE LINE is in effect, pressing the ENTER key while in the filearea will result in a new line being added.
When INPUTMODE OFF is in effect, pressing the ENTER key while in the filearea will result in the cursor moving to the beginning of the next line; scrolling the screen if necessary.
When INPUTMODE FULL is in effect, pressing the ENTER key while in the filearea will result in the cursor moving to the beginning of the next line; scrolling the screen if necessary.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

LINE

See Also:

INPUT

Status:

Incomplete. No support for FULL option.

SET INSERTMODE − put editor into or out of insert mode

Syntax:

[SET] INSERTMode *ON*|*OFF*|*TOGGLE*

Description:

The SET INSERTMODE command enable the user to set the insert mode within THE.
The *TOGGLE* option turns insert mode *ON* if it is currently *OFF* and vice versa.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

OFF

Status:

Complete.

SET INTERFACE − set overall behaviour of THE

Syntax:

[SET] INTerface *CLASSIC*|*CUA*

Description:

The SET INTERFACE command changes the behaviour of several operations within THE. THE normally operates in a block−mode manner, however many applications conform to the Common User Access (CUA) standard developed by IBM. This command specifies that CUA behaiour should occur on various actions during the edit session.
The major differences between CLASSIC and CUA behaviour, involve keyboard and mouse actions.
Various THE commands have CUA options to allow the user to customise the behaviour individual keys or the mouse to behave in a CUA manner.
Where behaviour is not related to particular key or mouse actions, this command provides the mechanism for changing the behaviour. The behaviour that SET INTERFACE affects:
− entering text in the filearea with a marked CUA block will
first delete the block and reposition the cursor
− executing SOS DELCHAR or SOS DELBACK will delete the
marked CUA block
− executing any positioning command, such as CURSOR DOWN,
FORWARD or CURSOR MOUSE, will unmark the CUA block

Compatibility:

XEDIT: N/A
KEDIT: Compatible with KEDIT for Windows.

Default:

CLASSIC

See Also:

MARK, CURSOR

Status:

Incomplete.

SET LINEFLAG − set the line characteristics of lines

Syntax:

[SET] LINEFLAG *CHAnge|NOCHange NEW|NONEW TAG|NOTAG [target]*

Description:

The SET LINEFLAGS command controls the line characteristics of lines in a file.

Each line in a file has certain characteristics associated with it depending on how the line has been modified. On reading a file from disk, all lines in the file are set to their default values.

Once a line is modified, or tagged, the characteristics of the line are set appropriately. A line that is added, is set to NEW; a line that is changed is set to CHANGE, and a line that is tagged with the TAG command, is set to TAG. All three characteristics can be on at the one time.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

NOCHANGE NONEW NOTAG

See Also:

TAG, SET HIGHLIGHT

Status:

Complete.

SET LINEND − allow/disallow multiple commands on command line

Syntax:

[SET] LINENd *ON|OFF [character]*

Description:

The SET LINEND command allows or disallows the execution of multiple commands on the command line . When setting LINEND ON, a *character* is specified as the LINEND character which delimits each command.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

OFF #

Status:

Complete.

SET MACRO − indicate if macros executed before commands

Syntax:

SET MACRO *ON|OFF*

Description:

The SET MACRO command allows the user to determine if macros are executed before a built−in command of the same name.

This command MUST be prefixed with SET to distinguish it from the MACRO command.

A macro with the same name as a built−in command will only be executed before the built−in command if SET IMPMACRO is ON, SET MACRO is ON, and the command was NOT executed with the COMMAND command.

Compatibility:

XEDIT: Compatible.

KEDIT: N/A

Default:

OFF

See Also:

MACRO, SET IMPMACRO, COMMAND

Status:

Complete.

SET MACROEXT − set default macro extension value

Syntax:

[SET] MACROExt *[ext]*

Description:

The SET MACROEXT command sets the value of the file extension to be used for macro files. When a macro file name is specified on the command line , a period '.' , then this value will be appended. If no value is specified for *ext* , then THE assumes that the supplied macro file name is the fully specified name for a macro.

The length of *ext* must be 10 characters or less.

The macro extension is only appended to a file if that file does not include any path specifiers.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

the

Status:

Complete.

SET MACROPATH − set default path for macro commands

Syntax:

[SET] MACROPath *PATH|path[s]*

Description:

The SET MACROPATH command sets up the search path from which macro command files are executed. Each directory is seperated by a colon (Unix) or semi−colon (DOS &OS/2). Only 20 directories are allowed to be specified.

When *PATH* is specified, the search path is set to the system PATH environment variable.

Compatibility:

XEDIT: N/A

KEDIT: Incompatible.

Default:

Path specified by env variable THE_MACRO_PATH

See Also:

MACRO, SET IMPMACRO

Status:

Complete.

SET MARGINS − set left and right margins for wordwrap

Syntax:

[SET] MARgins *left right [[+|−]indent]*

Description:

The SET MARGINS command sets the *left* and *right* margins and the number of columns to *indent* a paragraph.

These values are used with the SET WORDWRAP option.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

1 72 +0

See Also:

SET WORDWRAP, SET EQUIVCHAR

Status:

Complete.

SET MOUSE − turn mouse support on or off

Syntax:

[SET] MOUSE *ON|OFF*

Description:

The SET MOUSE command allows the user to turn on or off mouse support in THE. With mouse support, THE commands assigned to a mouse button event will be executed. See APPENDIX 3 for details on default mouse support.

If the platform does not support mouse operations, the default setting will be OFF.

Compatibility:

XEDIT: N/A

KEDIT: Compatible. Does not support all options.

Default:

ON − if mouse supported, OFF − otherwise

See Also:

DEFINE

Status:

Complete.

SET MSGLINE − set position and size of message line

Syntax:

[SET] MSGLine *ON M[+n|−n]|[+|−]n [lines] [Overlay]*

[SET] MSGLine *CLEAR*

Description:

The SET MSGLINE set command specifies the position of the message line and the size of the message line window.

The first form of positional parameters is:

M[+n|−n]

this sets the first line to be relative to the middle of
the screen. A positive value adds to the middle line number,
a negative subtracts from it.
eg. M+3 on a 24 line screen will be line 15
M−5 on a 24 line screen will be line 7

The second form of positional parameters is:

[+|−]n

this sets the first line to be relative to the top of the
screen (if positive or no sign) or relative to the bottom
of the screen if negative.
eg. +3 or 3 will set first line to line 3
−3 on a 24 line screen will set first line to line 21

If the resulting line is outside the bounds of the screen the position of the message line will become the middle line on the screen.

The *lines* argument specifies the maximum number of lines of error messages to display at the one time. If this value is specified as a whole number it must be less than or equal to the number of lines that could fit on the screen from the starting row. '*' can be specified to indicate that as many lines as possible should be displayed.

All options can be specified as the current EQUIVCHAR to retain the existing value.

The second format of the command clears the messages being displayed. This is useful in macros where you need to display an error message but also want to be able to clear it.

Compatibility:

XEDIT: Compatible.

The OVERLAY option is the default but ignored.

The second format is not supported.

KEDIT: Compatible

The OVERLAY option is the default but ignored.

The second format is not supported.

Default:

ON 2 5 Overlay

See Also:

SET EQUIVCHAR

Status:

Complete

SET MSGMODE − set display of messages on or off

Syntax:

[SET] MSGMode *ON|OFF*

Description:

The SET MSGMODE set command determines whether error messages will be displayed or suppressed.

Compatibility:

XEDIT: Does not support [Short|Long] options.

KEDIT: Compatible

Default:

ON

Status:

Complete

SET NEWLINES − set position of cursor after adding blank line

Syntax:

[SET] NEWLines *Aligned|Left*

Description:

The SET NEWLINES set command determines where the cursor displays after a new line is added to the file.

With *ALIGNED* , the cursor will display in the column of the new line immediately underneath the first non−blank character in the line above. With *LEFT* , the cursor will display in the first column of the new line.

Compatibility:

XEDIT: N/A

KEDIT: Same command, different functionality.

Default:

Aligned

Status:

Complete

SET NONDISP − specify character to display for non−displaying characters

Syntax:

[SET] NONDisp *character*

Description:

The SET NONDISP command allows the user to change the *character* that is displayed for non−displaying commands when SET ETMODE is OFF.

Compatibility:

XEDIT: Compatible.
KEDIT: N/A

Default:

#

See Also:

SET ETMODE

Status:

Complete.

SET NUMBER − turn prefix numbers on or off

Syntax:

[SET] NUMber *ON|OFF*

Description:

The SET NUMBER command allows the user to toggle the display of numbers in the prefix area .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

ON

See Also:

SET PREFIX

Status:

Complete.

SET PAGEWRAP − determine if page scrolling wraps at bottom/top of file

Syntax:

[SET] PAGEWRAP *ON|OFF*

Description:

The SET PAGEWRAP command allows the user to turn on or off the automatic wrapping of FORWARD and BACKWARD commands when the cursor is at the Bottom−of−File line or Top−of−File line respectively.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

ON

Status:

Complete.

SET PARSER − associates a language definition file with a parser

Syntax:

[SET] PARSER *parser file*

Description:

The SET PARSER defines a new syntax highlighting parser ; *parser* based on a language definition file; *file* .

The *file* is looked for in the directories specified by SET MACROPATH .

To specify one of the builtin parsers, prefix the filename with '*' . Therefore to define a parser called FRED using the builtin C parser, the command would be: SET PARSER FRED *C.TLD.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

SET COLORING, SET ECOLOUR, SET AUTOCOLOR, SET MACROPATH

Status:

Complete.

SET PENDING − set status of pending prefix commands

Syntax:

[SET] PENDing *ON string*

[SET] PENDing *OFF*

[SET] PENDing *BLOCK string*

Description:

The SET PENDING command allows the user to insert or remove commands from the pending prefix list.

ON string, simulates the user typing *string* in the prefix area of the focus line .

OFF, removes any pending prefix command from the focus line.

BLOCK string, simulates the user typing *string* in the PREFIX area of the focus line and identifies the prefix command to be a BLOCK command.

Compatibility:

XEDIT: Does not support ERROR option.

KEDIT: N/A

Status:

Complete.

SET POINT − assign a name to the current line

Syntax:

[SET] Point *.name [OFF]*

Description:

The SET POINT command assignes the specified name to the focus line , or removes the name from the line with the specified name. A valid line name must start with a '.' followed by alphanumeric characters. eg. .a .fred and .3AB are valid names.

When a line is moved within the same file, its line name stays with the line.

Compatibility:

XEDIT: Compatible. See below.

KEDIT: Compatible. See below.

Does not allow for multiple names for the same line.

Status:

Complete.

SET POSITION − determine if LINE/COL is displayed on idline

Syntax:

[SET] POSition *ON|OFF*

Description:

The SET POSITION command allows the user to turn on or off the display of LINE/COL on the idline .

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

ON

Status:

Complete.

SET PREFIX − set prefix area attributes

Syntax:

[SET] PREfix *ON [Left|Right] [n [m]]*
[SET] PREfix *Nulls [Left|Right] [n [m]]*
[SET] PREfix *OFF*
[SET] PREfix *Synonym newname oldname*

Description:

The first form of the SET PREFIX command allows the user to display the prefix area and optionally to select the position were the prefix should be displayed.
The second form of the SET PREFIX command is functionally the same as the first form. The difference is that when the prefix area is displayed with SET NUMBER ON, numbers are displyed with leading spaces rather than zeros; with SET NUMBER OFF, blanks are displayed instead of equal signs.
The third form, turns the display of the prefix area off. Executed from within the profile, the only effect is that the defaults for all files is changed. Executed from the command line, the SET PREFIX command changes the current window displays to reflect the required options.
The fourth form of the SET PREFIX command allows the user to specify a synonym for a prefix command or Rexx prefix macro. The *newname* is the command entered in the prefix area and *oldname* corresponds to an existing prefix command or a Rexx macro file in the MACROPATH ending in .the or whatever the value of SET MACROEXT is at the time the prefix command is executed. The *oldname* can also be the fully qualified filename of a Rexx macro.
The first and second forms of the SET PREFIX command allows the user to specify the width of the prefix area and optionally a gap between the prefix area and the filearea. *m* can be specified as an unsigned number between 2 and 20 inclusive. *n* can be specified as an unsigned number between 0 and 18, but less than the number specified in *m* .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.
Specification of prefix width is a THE−only option.

Default:

ON Left 6 0

Status:

Complete.

SET PRINTER − define printer spooler name

Syntax:

[SET] PRINTER *spooler|[OPTION options]*

Description:

The SET PRINTER command sets up the print spooler name to determine where output from the PRINT command goes.

The *options* can be one of the following: CPI n (characters per inch) LPI n (lines per inch) ORIENTation Portrait|Landscape FONT fontname (name of fixed width font)

No checking is done for printer options. ie. You may specify a font that THE doesn 't know about, and the printing process may not work after that.'

The defaults for page layout for Win32 are: CPI 16 LPI 8 ORIENTation Portrait FONT LinePrinter BM

options are only valid for Win32 platform. Printer output for the Win32 platform ALWAYS goes to the default printer. Therefore, the *spooler* option is invalid on this platform.

Compatibility:

XEDIT: N/A

KEDIT: Compatible. THE adds more funtionality.

Default:

LPT1 − DOS/OS2, lpr − Unix, default − Win32

See Also:

PRINT

Status:

Complete.

SET PSCREEN − set physical size of screen

Syntax:

[SET] PSCReen *height [width] [RESET|PRESET]*

Description:

The SET PSCREEN command allows the user to adjust the size of the physical screen to the size specified by *height* and *width* .

This command does not work on all platforms.

The optional argument [RESET|PRESET] are ignored; they are there for Kedit compatibility.

Compatibility:

XEDIT: N/A

KEDIT: Compatible. Ignores RESET|PRESET argument

Default:

System Dependent

Status:

Incomplete.

SET READONLY − allow/disallow changes to a file if it is readonly

Syntax:

[SET] READONLY *ON|OFF|FORCE*

Description:

The SET READONLY command allows the user to disallow changes to files if they are readonly. Normally, if a file is readonly, THE allows the user to make changes to the file contents while in the editing session, but does not allow the file to be saved.

With READONLY ON, THE disallows any changes to be made to the contents of the file in memory, in much the same way that THE disallows changes to be made to any files, if THE is started with the −r command line switch.

With READONLY FORCE, THE disallows any changes to be made to the contents of the file in memory, in the same way that THE disallows changes to be made to any files, if THE is started with the −r command line switch.

While the −r command line switch disallows changes to be made to any files, SET READONLY ON, only disallows changes to be made to readonly files. SET READONLY FORCE disallows changes to be made to the current file irrespective of whether it is readonly on disk.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

OFF

Status:

Complete.

SET REGEXP − specify the regular expression syntax to use

Syntax:

[SET] REGEXP *syntax*

Description:

The SET REGEXP command allows the user specify which of the many regular expression syntaxes to use when using regular expressions in a target . The *syntax* can be specified as one of:

EMACS
AWK
POSIX_AWK
GREP
EGREP
POSIX_EGREP
SED
POSIX_BASIC
POSIX_MINIMAL_BASIC
POSIX_EXTENDED
POSIX_MINIMAL_EXTENDED

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

EMACS

See Also:

LOCATE, ALL

Status:

Complete

SET REPROFILE − indicate if profile file to be executed for all files

Syntax:

[SET] REPROFile *ON|OFF*

Description:

The SET REPROFILE command allows the user to determine if the profile file is to reexecuted for files subsequenlty edited.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

OFF

See Also:

XEDIT, EDIT, THE

Status:

Complete

SET RESERVED − display a reserved line

Syntax:

[SET] RESERved *|+|−*n [colour] [text|OFF]*

Description:

The SET RESERVED command reserves a line for the display of arbitrary text by the user. The position is determined by +|−n. This number, if positive, specifies the line relative from the top of the display. A negative number is relative from the bottom of the display.

By specifying a line, say +3, then the third line from the top will be reserved, with the supplied text being displayed in that line.

The idline of a file will always be displayed after any reserved lines.

The status line is not considered part of the displayable area, so any positioning specifications ignore that line.

A reserved line can only be turned off by identifying it in the same way that it was defined. If a reserved line was added with the position specification of −1, it cannot be turned off with a position specification of 23, even though both position specifiers result in the same display line.

All reserved lines may be turned of by specifying * as the number of lines.

The colour option specifies the colours to use to display the reserved line. The format of this colour specifier is the same as for SET COLOUR . If no colour is specified, the colour of the reserved line will be the colour set by any SET COLOUR RESERVED command for the view or white on black by default.

The text of reserved lines can also included embedded control characters to control the colour of portions of the text. Assume the following SET CTLCHAR commands have been issued:

SET CTLCHAR ESCAPE !
SET CTLCHAR @ PROTECT BOLD RED ON WHITE
SET CTLCHAR % PROTECT GREEN ON BLACK

Then to display a reserved line using the specified colours:

SET RESERVED −1 normal!@bold red on white!%green on black

It is an error to try to reserve a line which is the same line as SET CURLINE .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

See Also:

SET COLOUR, SET CTLCHAR

Status:

Complete.

SET REXXOUTPUT − indicate where Rexx output is to go

Syntax:

[SET] REXXOUTput *File|Display n*

Description:

The SET REXXOUTPUT command indicates where output from the Rexx interpreter is to go; either captured to a file in the ring or displayed in a scrolling fashion on the screen.

Also specified is the maximum number of lines from the Rexx interpreter that are to be displayed or captured. This is particularly useful when a Rexx macro gets into an infinite loop.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

Display 1000

Status:

Complete.

SET SCALE − set position and status of scale line on screen

Syntax:

[SET] SCALe *ON|OFF [M[+n|−n]|[+|−]n]*

Description:

The SET SCALE command sets the position and status of the scale line for the current view.

The first form of parameters is:

M[+n|−n]

this sets the scale line to be relative to the middle of
the screen. A positive value adds to the middle line number,
a negative subtracts from it.

eg. M+3 on a 24 line screen will be line 15
M−5 on a 24 line screen will be line 7

The second form of parameters is:

[+|−]n

this sets the scale line to be relative to the top of the
screen (if positive or no sign) or relative to the bottom
of the screen if negative.

eg. +3 or 3 will set current line to line 3
−3 on a 24 line screen will be line 21

If the resulting line is outside the bounds of the screen the position of the current line will become the middle line on the screen.

It is an error to try to position the SCALE line on the same line as SET CURLINE .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

OFF M+1

Status:

Complete.

SET SCOPE − sets which lines are to be excluded from commands

Syntax:

[SET] SCOPE *All|Display*

Description:

The SET SCOPE command indicates whether lines not displayed as the result of a SET DISPLAY or ALL command are included in the scope of lines to be acted upon by other THE commands.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

Display

See Also:

SET DISPLAY, SET SELECT, ALL

Status:

Completed.

SET SCREEN − specify number of screens displayed

Syntax:

[SET] SCReen *n [Horizontal|Vertical]*

[SET] SCReen *Size l1|* [l2|*]*

Description:

The SET SCREEN command specifies the number of views of file(s) to display on screen at once. If the number of views specified is 2 and only one file is currently in the ring , two views of the same file are displayed.

The second form of SET SCREEN allows the user to specify the number of lines that each screen occupies. The sum of *l1* and *l2* must equal to lscreen.5 or lscreen.5 − 1 if the status line is displayed. The value of *l1* specifies the size of the topmost screen; *l2* specifies the size of the bottommost screen. Either *l1* or *l2* can be set to *, but not both. The * signifies that the screen size for the specified screen will be the remainder of the full display window after the size of the other screen has been subtracted.

The THE display can only be split into 1 or 2 screens.

Compatibility:

XEDIT: Does not support Width or Define options.

KEDIT: Does not support Split option.

A maximum of 2 screens are supported.

Default:

1

See Also:

SET STATUSLINE

Status:

Complete.

SET SELECT − sets the selection level for the specified lines

Syntax:

[SET] SELect *[+|−]n [target]*

Description:

The SET SELECT command sets the selection level for the indicated lines equal to *n* (if no signs are specified) or adds or subtracts n from the selection level currently set for the lines in the target.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

0

See Also:

SET SCOPE, SET DISPLAY, ALL

Status:

Complete.

SET SHADOW − determines if shadow lines are displayed or not

Syntax:

[SET] SHADOW *ON|OFF*

Description:

The SET SHADOW command indicates whether shadow line s are to be displayed.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

ON

See Also:

SET DISPLAY, SET SELECT, ALL

Status:

Completed.

SET SLK − set Soft Label Key definitions

Syntax:

[SET] SLK *n|OFF [text]*

Description:

The SET SLK command allows the user to specify a short text description to be displayed on the bottom of the screen, using the terminal 's built−in Soft Label Keys, or the last line of the screen.'

The *n* argument of the command represents the label number from left to right, with the first label numbered 1.

OFF turns off display of the Soft Label Keys. This is the same as executing [SET] SLK n with no optional text for each label displayed.

The main use for this command is to describe the function assigned to a function key, in place of a reserved line .

On those platforms that support a pointing device, clicking the left mouse button on the Soft Label Key, is equivalent to pressing the associated function key.

The number of Soft Label Keys displayed is dependent on which curses library THE is using. PDCurses can display 10 keys with the length of the *text* argument 7 characters on a screen that is 80 columns wide. The number of characters that can be displayed increases with the width of the screen. Other curses implementations, limit the number of Soft Label Keys to 8, with a text width of 8 characters. Some curses implementations do not support Soft Label Keys.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

OFF

See Also:

SET COLOUR

Status:

Complete.

SET SPAN − specify if a string target can span multiple lines (unavailable)

Syntax:

[SET] SPAN *ON|OFF [Blank|Noblank [n|*]]*

Description:

The SET SPAN set command determines if a character string that is the subject of a target search can span more than one line of the file.

Compatibility:

XEDIT: Compatible.

KEDIT: N/A

Default:

OFF Blank 2

Status:

Not started.

SET SPILL − specify if a string target can span multiple lines (unavailable)

Syntax:

[SET] SPILL *ON|OFF|WORD*

Description:

The SET SPILL set command determines how characters are spilt off the end of a line when the length of the line exceeds the truncation column.

Compatibility:

XEDIT: Compatible.

KEDIT: N/A

Default:

OFF

Status:

Not started.

SET STATOPT − set display options on statusline

Syntax:

[SET] STATOPT *ON option column [length [prompt]]*

[SET] STATOPT *OFF option|**

Description:

The SET STATOPT command allows the user to specify which internal settings of THE are to be displayed on the status line .

The *option* argument is any value returned by the EXTRACT command. eg NBFILE.1.

The syntax of the ON option, displays the specified value, at the position in the status line specified by *column* . If supplied, *length* specifies the number of characters, beginning at the first character of the returned value, to display. A value of 0 indicates that the full value if to be displayed. The optional *prompt* argument, allows the user to specify a string to display immediately before the returned value.

OFF, removes the specified *option* from displaying. If * is specified, all displayed options will be removed.

column is relative to the start of the status line . The value of *column* must be > 9, so that the version of THE is not obscured.

Options will be displayed in the order in which they are set.

If SET CLOCK or SET HEX are ON, these will take precedence over options specified with this command.

The more values you display the longer it will take THE to display the status line . Also, some values that are available via EXTRACT are not really suitable for use here. eg CURLINE.3.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

ON NBFILE.1 13 0 Files=

ON WIDTH.1 23 0 Width=

Status:

Complete.

SET STATUSLINE − set position of status line

Syntax:

[SET] STATUSLine *Top|Bottom|Off|GUI*

Description:

The SET STATUSLINE command determines the position of the status line for the editing session. TOP will place the status line on the first line of the screen; BOTTOM will place the status line on the last line of the screen; OFF turns off the display of the status line.

The GUI option is only meaningful for those platforms that support a separate status line window. If specified for non−GUI ports, the GUI option is equivalent to OFF.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Added GUI option for THEdit port.

Default:

Bottom

Status:

Complete

SET STAY − set condition of cursor position after CHANGE/LOCATE commands

Syntax:

[SET] STAY *ON|OFF*

Description:

The SET STAY set command determines what line is displayed as the current line after an unsuccessful LOCATE or successful CHANGE command.

With STAY ON, the current line remains where it currently is.

With STAY OFF, after an unsuccessful LOCATE , the current line becomes the Bottom−of−File line (or Top−of−File line if direction is backwards).

After a successful CHANGE , the current line is the last line affected by the CHANGE command.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

ON

Status:

Complete

SET SYNONYM − define synonyms for commands (unavailable)

Syntax:

[SET] SYNonym *ON|OFF*

[SET] SYNonym *[LINEND char] newname [n] definition*

Description:

The SET SYNONYM command allows the user to define synonyms for commands or macros.

The first format indicates if synonym processing is to be performed.

The second format defines a command synonym.

The synonym is *newname* , which effectively adds a new THE command with the definition specified by *definition* . The *n* parameter defines the minimum length of the abbreviation for the new command An optional LINEND character can be specified prior to *newname* if the *definition* contains multiple commands.

definition can be of the form: [REXX] command [args] [#command [args] [...]] (where # represents the LINEND character specified prior to *newname*)

If the optional keyword; 'REXX' , is supplied, the remainder of the command line is treated as a Rexx macro and is passed onto the Rexx interpreter (if you have one) for execution.

Only 1 level of synonym processing is carried out; therefore a synonym cannot be specified in the *definition* .

Compatibility:

XEDIT: Compatible. Does not support format that can reorder paramaters.

KEDIT: Compatible.

Default:

OFF

Status:

Incomplete.

SET TABKEY − set characteristics of the SOS TABF command

Syntax:

[SET] TABKey *Tab|Character Tab|Character*

Description:

The SET TABKEY sets the action to be taken when the SOS TABF command is executed.

Depending on the insert mode, the SOS TABF command will either display a raw tab character or will move to the next tab column.

The first operand refers to the behaviour of the SOS TABF command when SET INSERTMODE is OFF.

The second operand specifies the behaviour when the SOS TABF command is executed when SET INSERTMODE is ON.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

Tab Character

See Also:

SET EQUIVCHAR

Status:

Complete

SET TABLINE − set position and status of tab line on screen

Syntax:

[SET] TABLine *ON|OFF [M[+n|−n]|[+|−]n]*

Description:

The SET TABLINE command sets the position and status of the tab line for the current view.

The first form of parameters is:

M[+n|−n]

this sets the tab line to be relative to the middle of
the screen. A positive value adds to the middle line number,
a negative subtracts from it.
eg. M+3 on a 24 line screen will be line 15
M−5 on a 24 line screen will be line 7

The second form of parameters is:

[+|−]n

this sets the tab line to be relative to the top of the
screen (if positive or no sign) or relative to the bottom
of the screen if negative.
eg. +3 or 3 will set current line to line 3
−3 on a 24 line screen will be line 21

If the resulting line is outside the bounds of the screen the position of the current line will become the middle line on the screen.

It is an error to try to position the TABL line on the same line as SET CURLINE .

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

OFF −3

Status:

Complete.

SET TABS − set tab columns or tab length

Syntax:

[SET] TABS *n1 [n2 ... n32]*
[SET] TABS *INCR n*
[SET] TABS *OFF*

Description:

The SET TABS command determines the position of tab columns in THE.

The first format of SET TABS, specifies individual tab columns. Each column must be greater than the column to its left.

The second format specifies the tab increment to use. ie each tab column will be set at each *n* columns.

The third format specifies that no tab columns are to be set.

Tab columns are used by SOS TABF , SOS TABB and SOS SETTAB commands to position the cursor and also by the COMPRESS and EXPAND commands.

Compatibility:

XEDIT: Compatible. Does not support OFF option.
KEDIT: Compatible. Does not support OFF option.

Default:

INCR 8

Status:

Complete.

SET TABSIN − set tab processing on file input

Syntax:

[SET] TABSIn *ON|OFF [n]*

Description:

The SET TABSIN command determines if tabs read from a file are to be expanded to spaces and if so how many spaces.

Compatibility:

XEDIT: N/A
KEDIT: Does not support TABQUOTE option.

Default:

OFF 8

See Also:

SET TABSOUT

Status:

Complete.

SET TABSOUT − set tab processing on file output

Syntax:

[SET] TABSOut *ON|OFF [n]*

Description:

The SET TABSOUT command determines if spaces written to a file are to be compressed to tabs and if so how many spaces.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

OFF 8

See Also:

SET TABSIN

Status:

Complete.

SET TARGETSAVE − set type(s) of targets to save for subsequent LOCATEs

Syntax:

[SET] TARGETSAVE *ALL|NONE| STRING REGEXP ABSOLUTE RELATIVE POINT BLANK*

Description:

The SET TARGETSAVE command allows you to specify which target types are saved for subsequent calls to the LOCATE command without any parameters.

By default; SET TARGETSAVE ALL, the LOCATE command without any parameters, locates the last target irrespective of the type of target.

SET TARGETSAVE NONE turns off saving of targets, but does not delete any already saved target.

Any combination of the target types, STRING, REGEXP, ABSOLUTE, RELATIVE, POINT, or BLANK can be supplied. eg. SET TARGETSAVE STRING POINT.

As an example, having SET TARGETTYPE STRING then the only target saved will be one that has a string target component. ie. if you executed LOCATE /fred/ then LOCATE :3 then LOCATE, the final LOCATE will look for /fred/ NOT :3

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

ALL

See Also:

LOCATE

Status:

Complete.

SET THIGHLIGHT − specify if text highlighting is supported

Syntax:

[SET] THIGHlight *ON|OFF*

Description:

The SET THIGHLIGHT command allows the user to specify if a the result of a string LOCATE command should be highlighted. The colour that is used to highlight the found string is set by the THIGHLIGHT option of SET COLOUR . The found string is highlighted until a new line is added or deleted, a command is issued from the command line, another LOCATE or CLOCATE command is executed, a block is marked, or RESET THIGHLIGHT is executed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON − THE/KEDIT/KEDITW OFF − XEDIT/ISPF

See Also:

LOCATE, SET COLOUR

Status:

Complete.

SET TIMECHECK − specify if time stamp checking done

Syntax:

[SET] TIMECHECK *ON*|*OFF*

Description:

The SET TIMECHECK command allows the user to specify if a check is made of the modification time of the file being saved at the time of saving. This is done to alert the user if a file has changed since they began editing the file.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON

Status:

Complete.

SET TOFEOF − specify if TOF and BOF lines are displayed

Syntax:

[SET] TOFEOF *ON*|*OFF*

Description:

The SET TOFEOF command allows the user to specify if the Top−of−File line and the Bottom−of−File line are displayed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Default:

ON

Status:

Complete.

SET TRAILING − specify the truncation column

Syntax:

[SET] TRAILING *ON*|*OFF*|*SINGLE*|*EMPTY*

Description:

The SET TRAILING set command determines how trailing blanks on lines are handled when written to disk. TRAILING ON means that THE will not treat trailing blanks any differently from any other characters in the file. With TRAILING OFF, THE will remove trailing blanks when a file is read, remove them during an edit session, and not write any trailing blanks to the file. TRAILING SINGLE is the same as TRAILING OFF, except that a single blank character is appended to the end of every line when the file is written. TRAILING EMPTY is the same as TRAILING OFF, except that otherwise empty lines will be written with a single trailing blank.

Note that the default for this under THE is ON. This is because of the way that THE processes profile files. If the default was OFF, and you had TRAILING ON in your profile, then there would be no way to retain the original trailing blanks.

Compatibility:

XEDIT: Compatible.
KEDIT: N/A

Default:

ON

Status:

Complete. Some trailing blank behaviour while editing files incomplete.

SET TRUNC − specify the truncation column

Syntax:

[SET] TRunc *n*|*

Description:

The SET TRUNC set command determines the truncation column. This is the rightmost column of text upon which THE commands are effective.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

Default:

*

Status:

Incomplete.

SET TYPEAHEAD − set behaviour of screen redraw

Syntax:

[SET] TYPEAhead *ON*|*OFF*

Description:

The SET TYPEAHEAD set command determines whether or not THE uses the curses screen display optimization techniques.
With TYPEAHEAD ON, curses will abort screen display if a keystroke is pending.
With TYPEAHEAD OFF, curses will not abort screen display if a keystroke is pending.
For BSD based curses, this function has no effect.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Default:

OFF

Status:

Complete.

SET UNDOING − turn on or off undo facility for the current file

Syntax:

[SET] UNDOING *ON*|*OFF*

Description:

The SET UNDOING command allows the user to turn on or off the undo facility for the current file.
At this stage in the development of THE, setting UNDOING to OFF stops THE from saving changes made to lines in a file, and prevents those lines from being able to be RECOVERed.
Setting UNDOING to OFF will increase the speed at which THE can execute CHANGE and DELETE commands.

Compatibility:

XEDIT: N/A
KEDIT: Does not support optional arguments.

Default:

ON

Status:

Complete.

SET UNTAA − specifies if "Unsigned Numerical Targets Are Absolute"

Syntax:

[SET] UNTAA *ON|OFF*

Description:

The SET UNTAA command allows the user to turn on or off the behaviour of unsigned numerical targets.

Numerical targets have the form [:|;|+|−]nn. By default, if the optional portion of the target is not supplied, then a '+' is assumed. WIth SET UNTAA set to ON, if the optional portion of the target is not supplied, then a ':' is assumed.

Caution: This SET command affects all numerical targets, not just targets in the LOCATE command.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

OFF

Status:

Complete.

SET VERIFY − set column display limits

Syntax:

[SET] Verify *first [last]*

Description:

The SET VERIFY command sets the column limits for the display of the current file. *first* specifies the first column to be displayed and *last* specifies the last column to be displayed.

If no *last* option is specified '*' is assumed.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: Does not implement HEX display nor multiple column pairs.

KEDIT: Does not implement HEX display nor multiple column pairs.

Default:

1 *

See Also:

SET ZONE, SET EQUIVCHAR

Status:

Complete.

SET WIDTH − set width of maximum line that THE can edit

Syntax:

[SET] WIDTH *n*

Description:

The SET WIDTH command specifies the maximum length that a line can be within the edit session. This command is effectively the same as the −w command line switch.

The value *n* MUST be between 10 and 32700.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

512

SET UNTAA − specifies if "Unsigned Numerical Targets Are Absolute" 88

Status:

Complete.

SET WORD − controls what THE considers a word to be

Syntax:

[SET] WORD *NONBlank|ALPHAnum*

Description:

The SET WORD set command determines what sequence of characters THE considers a word to be. This is used in command such as SOS DELWORD , SOS TABWORDF and MARK WORD to specify the boundaries of the word.

The default setting for SET WORD is *NONBlank* . THE treats all sequences of characters seperated by a blank (ASCII 32) as words.

With *ALPHAnum* THE treats a group of consecutive alphanumeric characters as a word. THE also includes the underscore character and characters with an ASCII value > 128 as alphanumeric.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

NONBlank

Status:

Complete.

SET WORDWRAP − set wordwrap feature on or off

Syntax:

[SET] WORDWrap *ON|OFF*

Description:

The SET WORDWRAP set command determines whether wordwrap occurs when the cursor moves past the right margin (as set by the SET MARGINS command).

With WORDWRAP ON, the line, from the beginning of the word that exceeds the right margin, is wrapped onto the next line. The cursor position stays in the same position relative to the current word.

With WORDWRAP OFF, no word wrap occurs.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

OFF

See Also:

SET MARGINS

Status:

Complete.

SET WRAP − enable/disable string locates around the end of the file

Syntax:

[SET] WRap *ON|OFF*

Description:

The SET WRAP set command determines whether THE will look for a string target off the ends of the file.

With WRAP OFF, THE will attempt to locate a string target from the current line to the end of file (or top of file if the locate is a backwards search).

With WRAP ON, THE will attempt to locate a string target from the current line to the end of file (or top of file if the locate is a backwars search) and wrap around the end of the file and continue searching until the current line is reached.

If the string target is located after wrapping around the end of the file, the message 'Wrapped...' is displayed.

Commands affected by SET WRAP are; LOCATE , FIND , NFIND , FINDUP and NFINDUP .

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

Default:

OFF

See Also:

LOCATE, FIND, NFIND, FINDUP, NFINDUP

Status:

Complete.

SET XTERMINAL − set X terminal to execute under X

Syntax:

[SET] XTERMinal *program*

Description:

The SET XTERMINAL set command allows the user to specify the full quallified file name of the program to run when the OS , DOS or ! command is entered without parameters when running the X version of THE.

Compatibility:

XEDIT: N/A

KEDIT: N/A

Default:

System dependent but usually one of:

Status:

Complete.

SET ZONE − set column limits for editing

Syntax:

[SET] Zone *first [last]*

Description:

The SET ZONE command sets the column limits for various other editor commands, such as LOCATE and CHANGE . It effectively restricts to the specified columns those parts of the file which can be acted upon.

If no *last* option is specified '*' is assumed.

All options can be specified as the current EQUIVCHAR to retain the existing value.

Compatibility:

XEDIT: Compatible.

KEDIT: Compatible.

Default:

1 *

See Also:

SET VERIFY, SET EQUIVCHAR

Status:

Complete.

Return to Table of Contents

SOS COMMAND REFERENCE

SOS ADDLINE − add blank line after focus line

Syntax:

SOS ADDline

Description:

The SOS ADDLINE command inserts a blank line in the file following the focus line. The cursor is placed in the column under the first non−blank in the focus line.

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

SOS LINEADD, SOS DELLINE

Status:

Complete

SOS BLOCKEND − move cursor to end of marked block

Syntax:

SOS BLOCKEnd

Description:

The SOS BLOCKEND command moves the cursor to the ending line and column of the marked block. If the cursor is on the command line, the last line of the marked block becomes the current line. If no marked block is in the current file, an error is displayed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS BLOCKSTART

Status:

Complete.

SOS BLOCKSTART − move cursor to start of marked block

Syntax:

SOS BLOCKStart

Description:

The SOS BLOCKSTART command moves the cursor to the starting line and column of the marked block. If the cursor is on the command line, the first line of the marked block becomes the current line.

If no marked block is in the current file, an error is displayed.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS BLOCKEND

Status:

Complete.

SOS BOTTOMEDGE − move cursor to bottom edge of FILEAREA

Syntax:

SOS BOTTOMEdge

Description:

The SOS BOTTOMEDGE command moves the cursor to the last enterable line in the filearea or prefix area . If the cursor is on the command line, the cursor moves to the first enterable line of the filearea .

Compatibility:

XEDIT: N/A
KEDIT: Comaptible.

See Also:

SOS TOPEDGE

Status:

Complete.

SOS CUADELBACK − delete the character to the left of the cursor

Syntax:

SOS CUADELBAck

Description:

The SOS CUADELBACK command deletes the character to the right of the current cursor position. It differs from SOS DELBACK in the case when the cursor is in the first column of the file and in the FILEAREA. Then, the cursor first moves to the last character of the previous line, and deletes this character.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SOS DELBACK, SOS CUADELCHAR

Status:

Complete

SOS CUADELCHAR − delete character under cursor

Syntax:

SOS CUADELChar

Description:

The SOS CUADELCHAR command deletes the character under the cursor. Text to the right is shifted to the left. It differs from SOS DELCHAR in the case when the cursor is after the last character of the line and in the FILEAREA. Then, the next line is joined with the current line.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SOS CURDELBACK, SOS DELCHAR

Status:

Complete

SOS CURRENT − move cursor to current line

Syntax:

SOS CURRent

Description:

The SOS CURRENT command moves the cursor to the current column of the cursor line from any window.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete

SOS CURSORADJ − move first non−blank character to cursor

Syntax:

SOS CURSORAdj

Description:

The SOS CURSORADJ command moves text in the focus line so that the first non−blank character appears under the cursor position.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS CURSORSHIFT

Status:

Complete

SOS CURSORSHIFT − move text to right of cursor to cursor

Syntax:

SOS CURSORSHIFT

Description:

The SOS CURSORSHIFT command moves text in the focus line so that the first non−blank character to the right of the cursor is shifted to under the cursor position.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SOS CURSORADJ

Status:

Complete

SOS DELBACK − delete the character to the left of the cursor

Syntax:

SOS DELBAck

Description:

The SOS DELBACK command moves the cursor one character to the left and deletes the character now under the cursor.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS DELCHAR, SOS CUADELCHAR

Status:

Complete

SOS DELCHAR − delete character under cursor

Syntax:

SOS DELChar

Description:

The SOS DELCHAR command deletes the character under the cursor. Text to the right is shifted to the left.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS DELBACK

Status:

Complete

SOS DELEND − delete to end of line

Syntax:

SOS DELEnd

Description:

The SOS DELEND command deletes all characters from the current column to the end of line.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete.

SOS DELLINE − delete focus line

Syntax:

SOS DELLine

Description:

The SOS DELLINE command deletes the focus line .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

SOS LINEDEL, SOS ADDLINE

Status:

Complete

SOS DELWORD − delete word at or right of cursor

Syntax:

SOS DELWord

Description:

The SOS DELWORD command deletes the word at or to the right of the current cursor position and any spaces following the word.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete

SOS DOPREFIX − execute any pending prefix commands

Syntax:

SOS DOPREfix

Description:

The SOS DOPREFIX command executes any pending prefix commands.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete

SOS EDIT − edit a file from directory list

Syntax:

SOS EDIT

Description:

The SOS EDIT command allows the user to edit a file, chosen from a directory list (the file DIR.DIR).

Compatibility:

XEDIT: N/A
KEDIT: Compatible with default definition for Alt−X key.

Status:

Complete.

SOS ENDCHAR − move cursor to end of focus line

Syntax:

SOS ENDChar

Description:

The SOS ENDCHAR command moves the cursor to the position after the last character displayed in the current window.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS STARTENDCHAR

Status:

Complete.

SOS EXECUTE − move cursor to command line and execute command

Syntax:

SOS EXecute

Description:

The SOS EXECUTE command moves the cursor to the command line and executes any command that is displayed there.

Compatibility:

XEDIT: N/A
KEDIT: Compatible

Status:

Complete.

SOS FIRSTCHAR − move cursor to first non−blank of field

Syntax:

SOS FIRSTCHar

Description:

The SOS FIRSTCHAR command moves the cursor to the first non−blank character of the cursor field

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS FIRSTCOL

Status:

Complete.

SOS FIRSTCOL − move cursor to first column of field

Syntax:

SOS FIRSTCOl

Description:

The SOS FIRSTCOL command moves the cursor to the first column of the cursor field .

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS FIRSTCHAR, SOS LASTCOL

Status:

Complete.

SOS INSTAB − shift text to next tab column

Syntax:

SOS INSTAB

Description:

The SOS INSTAB command shifts all text from the current cursor position in the filearea to the next tab column.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SET TABS

Status:

Complete.

SOS LASTCOL − move cursor to last column of field

Syntax:

SOS LASTCOl

Description:

The SOS LASTCOL command moves the cursor to the last column of the cursor field .

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SOS FIRSTCOL

Status:

Complete.

SOS LEFTEDGE − move cursor to left edge of window

Syntax:

SOS LEFTEdge

Description:

The SOS LEFTEDGE command moves the cursor to the leftmost edge of the filearea if not on the command line or to the leftmost edge of the command line if on the command line .

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS RIGHTEDGE, SOS PREFIX

Status:

Complete.

SOS LINEADD − add blank line after focus line

Syntax:

SOS LINEAdd

Description:

The SOS LINEADD command inserts a blank line in the file following the focus line . The cursor is placed in the column under the first non−blank in the focus line .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

SOS ADDLINE, SOS LINEDEL

Status:

Complete

SOS LINEDEL − delete focus line

Syntax:

SOS LINEDel

Description:

The SOS LINEDEL command deletes the focus line .

Compatibility:

XEDIT: Compatible.
KEDIT: Compatible.

See Also:

SOS DELLINE, SOS LINEADD

Status:

Complete

SOS MAKECURR − make focus line the current line

Syntax:

SOS MAKECURR

Description:

The SOS MAKECURR command set the current line to the focus line .

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete

SOS MARGINL − move cursor to the left margin column

Syntax:

SOS MARGINL

Description:

The SOS MARGINL command moves the cursor to the left margin column.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.
Although, when issued from the command line, nothing happens.

See Also:

SOS MARGINR

Status:

Complete

SOS MARGINR − move cursor to the right margin column

Syntax:

SOS MARGINR

Description:

The SOS MARGINR command moves the cursor to the right margin column.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.
Although, when issued from the command line, nothing happens.

See Also:

SOS MARGINL

Status:

Complete

SOS PARINDENT − move cursor to the paragraph indent column

Syntax:

SOS PARINDent

Description:

The SOS PARINDENT command moves the cursor to the paragraph indent column.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.
Although, when issued from the command line, nothing

happens.

Status:

Complete

SOS PASTECMDLINE − copy contents of marked block to command line

Syntax:

SOS PASTECMDline

Description:

The SOS PASTECMDLINE command copies the contents of the marked block to the command line at the current cursor location.

Marked blocks that span one line only are allowed to be pasted.

Compatibility:

XEDIT: N/A
KEDIT: N/A

Status:

Complete

SOS PREFIX − move cursor to leftmost edge of prefix area

Syntax:

SOS PREfix

Description:

The SOS PREFIX command moves the cursor to the rightmost edge of the prefix area .

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS LEFTEDGE, SOS RIGHTEDGE

Status:

Complete.

SOS QCMND − move cursor to command line and clear

Syntax:

SOS QCmnd

Description:

The SOS QCMND command moves the cursor to the first column of the command line and clears it.

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS EXECUTE

Status:

Complete.

SOS RIGHTEDGE − move cursor to right edge of window

Syntax:

SOS RIGHTEdge

Description:

The SOS RIGHTEDGE command moves the cursor to the rightmost edge of the filearea if not on the command line or to the rightmost edge of the command line if on the command line .

Compatibility:

XEDIT: N/A
KEDIT: Compatible

See Also:

SOS LEFTEDGE, SOS PREFIX

Status:

Complete.

SOS SETTAB − set a tab column at the cursor position

Syntax:

SOS SETTAB

Description:

The SOS SETTAB command sets a tab column at the position of the cursor in the filearea . This command is ignored if issued elsewhere. If a tab column is already set at the cursor position, the tab column is cleared.

Compatibility:

XEDIT: N/A
KEDIT: Kedit does not toggle the tab column, but only sets it.

See Also:

SET TABS

Status:

Complete.

SOS STARTENDCHAR − move cursor to end/start of focus line

Syntax:

SOS STARTENDChar

Description:

The SOS STARTENDCHAR command moves the cursor to the first character displayed in the cursor field , if the cursor is after the last character displayed in the cursor field , or to the position after the last character displayed in the cursor field , if the cursor is anywhere else.

Compatibility:

XEDIT: N/A
KEDIT: N/A

See Also:

SOS ENDCHAR

Status:

Complete.

SOS TABB − move cursor to previous tab stop

Syntax:

SOS TABB

Description:

The SOS TABB command causes the cursor to move to the previous tab column as set by the SET TABS command. If the resulting column is beyond the left hand edge of the main window, the window will scroll half a window.

Compatibility:

XEDIT: Does not allow arguments.
KEDIT: Compatible. See below.
Does not line tab to next line if before the left hand tab column.

See Also:

SET TABS, SOS TABF

Status:

Complete.

SOS TABF − move cursor to next tab stop

Syntax:

SOS TABf

Description:

The SOS TABF command causes the cursor to move to the next tab column as set by the SET TABS command. If the resulting column is beyond the right hand edge of the main window, the window will scroll half a window.

Compatibility:

XEDIT: Does not allow arguments.
KEDIT: Compatible. See below.
Does not line tab to next line if after the right hand tab column.

See Also:

SET TABS, SOS TABB

Status:

Complete.

SOS TABFIELDB − move cursor to previous enterable field

Syntax:

SOS TABFIELDB

Description:

The SOS TABFIELDB command causes the cursor to move to the first column of the current enterable field. If the cursor is already in the first column of the current field the cursor moves to the first column of the previous enterable field on the screen. This command is intended to mimic the behaviour of the SHIFT−TAB key on a 3270 terminal.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS TABFIELDF

Status:

Complete.

SOS TABFIELDF − move cursor to next enterable field

Syntax:

SOS TABFIELDf

Description:

The SOS TABFIELDF command causes the cursor to move to the next enterable field on the screen. This command is intended to mimic the behaviour of the TAB key on a 3270 terminal.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

See Also:

SOS TABFIELDB

Status:

Complete.

SOS TABWORDB − move cursor to beginning of previous word

Syntax:

SOS TABWORDB

Description:

The SOS TABWORDB command causes the cursor to move to the first character of the word to the left or to the start of the line if no more words precede. If the resulting column is beyond the left hand edge of the filearea , the window will scroll half a window.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

SOS TABWORDF

Status:

Complete.

SOS TABWORDF − move cursor to start of next word

Syntax:

SOS TABWORDf

Description:

The SOS TABWORDF command causes the cursor to move to the first character of the next word to the right or to the end of the line if no more words follow. If the resulting column is beyond the right hand edge of the filearea , the window will scroll half a window.

Compatibility:

XEDIT: N/A

KEDIT: Compatible.

See Also:

SOS TABWORDB

Status:

Complete.

SOS TOPEDGE − move cursor to top edge of filearea

Syntax:

SOS TOPEdge

Description:

The SOS TOPEDGE command moves the cursor to the first enterable line in the filearea or prefix area . If the cursor is on the command line , the cursor moves to the first enterable line of the filearea .

Compatibility:

XEDIT: N/A

KEDIT: Comaptible.

See Also:

SOS BOTTOMEDGE

Status:

Complete.

SOS UNDO − undo changes to the current line

Syntax:

SOS UNDO

Description:

The SOS UNDO command causes the contents of the focus line (or the command line) to be reset to the contents before the cursor was positioned there.

Compatibility:

XEDIT: N/A
KEDIT: Compatible.

Status:

Complete.

Return to Table of Contents

QUERY, EXTRACT and STATUS

The following lists the valid variables that can be queried and also shows what values are returned. For both QUERY and STATUS , the values are concatenated together and displayed as a single line. For EXTRACT the REXX variables that are set are defined. The capitalised portion of the variable is the minimum abbreviation recognised.
The bracketed text at the end of the description indicates from which commands a valid response will be supplied.
(Q−Query, E−Extract, M−Modify and S−Status).

ALT

The number of alterations to the current file since the last SAVE or automatic save via SET AUTOSAVE . Can be set by SET ALT
(QES)

alt.0 − 2

alt.1 − Number of alterations since last SAVE or autosave

alt.2 − Number of alterations since last SAVE

ARBchar

The status of SET ARBCHAR and the ARBCHAR characters.
(QEMS)

arbchar.0 − 3

arbchar.1 − ON|OFF

arbchar.2 − multiple match character

arbchar.3 − single match character

AUTOCOLOR [*|mask]
AUTOCOLOUR [*|mask]

The status of auto colouring for the supplied file mask or "magic number". Set by SET AUTOCOLOR or SET AUTOCOLOUR . The variable name is spelt the save way that the option is specified.
(QES)

autocolor.0 − 3

autocolor.1 − filemask or "magic number"

autocolor.2 − name of parser associated or "NULL"

autocolor.3 − MAGIC (if "magic number") or empty

If no file mask or "magic number" is supplied as a parameter or * is passed, details of all parser mappings are set as follows:

autocolor.0 − number of parser mappings

autocolor.1 − first mapping (mask parser [MAGIC])

autocolor.n − nth mapping

AUtosave

The status of SET AUTOSAVE and/or the frequency setting.
(QEMS)

autosave.0 − 1

autosave.1 − OFF|n

AUTOSCroll

The status of SET AUTOSCROLL and/or the frequency setting.
(QEMS)

autoscroll.0 − 1

autoscroll.1 − OFF|HALF|n

BACKup

Indicates if a .bak file is kept after editing.
(QEMS)

backup.0 − 1

backup.1 − ON|OFF|TEMP|KEEP|INPLACE

BEEP

Indicates if the bell is sounded on display of error messages. Set by SET BEEP
(QEMS)

beep.0 − 1

beep.1 − ON|OFF

BLOCK

Returns information about the marked block , if any.
(QE)

block.0 − 6 if a marked block exists, or 1 for NONE

block.1 − type of marked block (LINE|BOX|NONE|WORD|COLUMN|CUA)

block.2 − line number of start of block

block.3 − column number of start of block

block.4 − line number of end of block

block.5 − column number of end of block

block.6 − file name containing marked block

CASE

The settings related to the CASE of characters entered, searched for, changed and sorted. Set by SET CASE
(QEMS)

case.0 − 4

case.1 − MIXED|UPPER|LOWER

case.2 − RESPECT|IGNORE

case.3 − RESPECT|IGNORE

case.4 − RESPECT|IGNORE

CLEARErrorkey

Returns the key that clears the message line . If no specific key is defined, ie ALL keys perform a reset, then * is returned. Set by SET CLEARERRORKEY .
(QEMS)

clearerrorkey.0 − 1

clearerrorkey.1 − keyname|*

CLEARScreen

Indicates if the screen is cleared on exit from THE. Set by SET CLEARSCREEN .
(QEMS)

clearscreen.0 − 1

clearscreen.1 − ON|OFF

CLOCK

Indicates if the time is displayed on the status line. Set by SET CLOCK .
(QEMS)

clock.0 − 1

clock.1 − ON|OFF

CMDArrows

Returns the settings for arrow key behaviour on command line . Set by SET CMDARROWS .
(QEMS)

cmdarrows.0 − 1

cmdarrows.1 − RETRIEVE|TAB

CMDline

The settings for the command line . Set by SET CMDLINE .
(QEMS)
If CMDLINE OFF

cmdline.0 − 1

cmdline.1 − OFF

If CMDLINE ON, BOTTOM or TOP

cmdline.0 − 3

cmdline.1 − ON|TOP|BOTTOM

cmdline.2 * − line number within window of command line

cmdline.3 * − contents of command line

* these values are only returned with EXTRACT

COLOR [*|area]

COLOUR [*|area]

Displays the current color settings for the file. Set by SET COLOR or SET COLOUR . The variable name is spelt the save way that the option is specified.
With the [*] option, (or no option), returns color settings for all areas of the screen.
(QE)

color.0 − 28

color.1 − FILEAREA followed by its color

color.2 − CURLINE followed by its color

color.3 − BLOCK followed by its color

color.4 − CBLOCK followed by its color

color.5 − CMDLINE followed by its color

color.6 − IDLINE followed by its color

color.7 − MSGLINE followed by its color

color.8 − ARROW followed by its color

color.9 − PREFIX followed by its color

color.10 − CPREFIX followed by its color

color.11 − PENDING followed by its color

color.12 − SCALE followed by its color

color.13 − TOFEOF followed by its color

color.14 − CTOFEOF followed by its color

color.15 − TABLINE followed by its color

color.16 − SHADOW followed by its color

color.17 − STATAREA followed by its color

color.18 − DIVIDER followed by its color

color.19 − RESERVED followed by its color

color.20 − NONDISP followed by its color

color.21 − HIGHLIGHT followed by its color

color.22 − CHIGHLIGHT followed by its color

color.23 − SLK followed by its color

color.24 − GAP followed by its color

color.25 − CGAP followed by its color

color.26 − ALERT followed by its color

color.27 − DIALOG followed by its color

color.28 − BOUNDMARKER followed by its color

color.29 − FILETABS followed by its color

With the [*area*] option, returns color settings for the specified area of the screen.
(QE)

color.0 − 1

color.1 − *area* followed by its color

COLORING
COLOURING

Displays the current color settings for the file. The variable name is spelt the same way that the option is specified. Set by SET COLORING or SET COLOURING .
(QEMS)

coloring.0 − 3

coloring.1 − ON|OFF

coloring.2 − AUTO|parser (if coloring.1 is ON)

coloring.3 − parser (if coloring.1 is ON)

COLUMN

Displays the current value of the focus column.
(QE)

column.0 − 1

column.1 − Column number of focus column

COMPAT

The settings for the compatibility mode. Set by SET COMPAT .
(QEMS)

compat.0 − 3

compat.1 − THE|XEDIT|KEDIT|KEDITW|ISPF (compatibility LOOK)

compat.2 − THE|XEDIT|KEDIT|KEDITW|ISPF (compatibility FEEL)

compat.3 − THE|XEDIT|KEDIT|KEDITW|ISPF (compatibility KEYS)

CTLchar [*|char]

The definitions of control characters that affect the display of reserved line s. Set by SET CTLCHAR .
(QES)
With the [*] option, (or no option), returns a list of all control characters that have been defined.

ctlchar.0 − 3

ctlchar.1 − ON|OFF

ctlchar.2 − The character defined as the escape character.

ctlchar.3 − List of defined control characters, if any.

With the [*char*] option, returns color settings for the specified control character.

ctlchar.0 − 1 if ctlchar.1 is OFF, otherwise 2

ctlchar.1 − PROTECT|NOPROTECT|OFF

ctlchar.2 − The color defined for this control character.

CURline

The value and position of the current line within the current file. Set by SET CURLINE . Also provides the contents of the focus line and an indication of the status of the focus line for the current session.
(QEMS)
With EXTRACT command:

curline.0 − 6

curline.1 − curline setting

curline.2 − line number within window of current line

curline.3 − contents of focus line

curline.4 − ON|OFF (ON if line has changed or been added this session)

curline.5 − OLD|OLD CHANGED|NEW CHANGED

curline.6 − selection level of focus line (same as select.1)

Note: If SET COMPAT (feel) is set to XEDIT, curline.3 will always return the contents of the current line .
The setting of curline.5 is as follows: OLD: The line existed in its current form in the file before THE began. OLD CHANGED: The line was in the file before THE started, but has been changed during the current editing session. NEW CHANGED: The line has been added to the file during the current editing session.
With QUERY , MODIFY and STATUS commands:

curline.0 − 1

curline.1 − curline setting

CURSor

The position of the cursor within the displayed screen and file at the time the EXTRACT /CURSOR/ is issued and the position of the cursor at the time the macro was issued. If the cursor is not in the filearea , then line and column values will be set to −1.
(QE)

cursor.0 − 8

cursor.1 − line number of cursor in screen (current)

cursor.2 − column number of cursor in screen (current)

cursor.3 − line number of cursor in file (current)

cursor.4 − column number of cursor in file (current)

cursor.5 − line number of cursor in screen (original)

cursor.6 − column number of cursor in screen (original)

cursor.7 − line number of cursor in file (original)

cursor.8 − column number of cursor in file (original)

CURSORSTay

The setting for the behaviour of the cursor after a file scroll. Set by SET CURSORSTAY .
(QEMS)

cursorstay.0 − 1

cursorstay.1 − ON|OFF

DEFINE [key|mousedef IN window|*]

Returns details about the commands associated with a keyboard key or mouse key. The details returned are the same as those displayed by the SHOWKEY command. Set by DEFINE .
(QEM)
Three forms are available:
With no parameter or * , all details about all key and mouse definitions are returned. (Similar to SHOWKEY all)

define.0 − number of all definitions

define.1 − command assigned to the 1st key or mouse

define.n − command assigned to the nth key or mouse

With a *key* mnemonic specified, details about this one key are returned.

define.0 − 1

define.1 − command assigned to the key

With a *mousedef IN window* specified, details about this one mouse mapping for a window are returned.

define.0 − 1

define.1 − command assigned to the mouse for the window

DEFSORT

Specifies the order in which files are sorted in the DIR.DIR file. Set by SET DEFSORT .
(QEMS)

defsort.0 − 2

defsort.1 − DIRECTORY|NAME|SIZE|TIME|DATE|OFF

defsort.2 − ASCENDING|DESCENDING

DIRFILEID

The value of the path and filename of the focus line in a DIR.DIR file.
(E)

dirfileid.0 − 2

dirfileid.1 − full path of directory

dirfileid.2 − file name at focus line

DIRInclude

The value of the file type masks currently in place for display in DIR.DIR file. Set by SET DIRINCLUDE .
(QEMS)

dirinclude.0 − 1

dirinclude.1 − list of file types shown in DIR.DIR file

DISPlay

Indicates the range of selection levels current. Set by SET DISPLAY .
(QEMS)

display.0 − 2

display.1 − display low value

display.2 − display high value

ECOLOR [*|category]

ECOLOUR [*|category]

Displays the current color settings for syntax highlighting. Set by SET ECOLOR or SET ECOLOUR .
The variable name is spelt the save way that the option is specified.
With the [*] option, (or no option), returns color settings for all categories.
(QE)

ecolor.0 − 35

ecolor.1 − A followed by its color

ecolor.2 − B followed by its color

ecolor.3 − C followed by its color

ecolor.4 − D followed by its color

ecolor.5 − E followed by its color

ecolor.6 − F followed by its color

ecolor.7 − G followed by its color

ecolor.8 − H followed by its color

ecolor.9 − I followed by its color

ecolor.10 − J followed by its color

ecolor.11 − K followed by its color

ecolor.12 − L followed by its color

ecolor.13 − M followed by its color

ecolor.14 − N followed by its color

ecolor.15 − O followed by its color

ecolor.16 − P followed by its color

ecolor.17 − Q followed by its color

ecolor.18 − R followed by its color

ecolor.19 − S followed by its color

ecolor.20 − T followed by its color

ecolor.21 − U followed by its color

ecolor.22 − V followed by its color

ecolor.23 − W followed by its color

ecolor.24 − X followed by its color

ecolor.25 − Y followed by its color

ecolor.26 − Z followed by its color

ecolor.27 − 1 followed by its color

ecolor.28 − 2 followed by its color

ecolor.29 − 3 followed by its color

ecolor.30 − 4 followed by its color

ecolor.31 − 5 followed by its color

ecolor.32 − 6 followed by its color

ecolor.33 − 7 followed by its color

ecolor.34 − 8 followed by its color

ecolor.35 − 9 followed by its color

With the [*category*] option, returns color settings for the specified syntax highlighting category.

(E)

color.0 − 1

color.1 − [*category*] followed by its color

EFILEId

The original full filename of the current file. See SET FILENAME for a full description of the components of a file name. EFILEID.2 differs from the value returned by KEDIT.

(QE)

efileid.0 − 2

efileid.1 − Original full file name.

efileid.2 − Original file name used to edit the file.

EOF

Indicates if the current line is on the Bottom−of−File line .
(QES)

eof.0 − 1

eof.1 − ON|OFF

EOLout

Returns the value of the end−of−line character(s).
(QEMS)

eolout.0 − 1

eolout.1 − LF|CRLF|CR|NONE

EQUIVChar

Returns the equivalence character. Set by SET EQUIVCHAR .
(QEMS)

equivchar.0 − 2

equivchar.1 − equivalence character

ETMODE

Indicates if extended display mode is set. Set by SET ETMODE .
(QEMS)

etmode.0 − 2

etmode.1 − ON|OFF

etmode.2 − character ranges if not all ON or OFF

FExt

The extension of the current file. The characters following the trailing . character. Same as FType.
(QEM)

fext.0 − 1

fext.1 − File extension.

FIELD

Details about the current cursor field.
(QE)

field.0 − 4

field.1 − contents of the cursor field

field.2 − character under the cursor

field.3 − column position in the cursor field

field.4 − COMMAND|TEXT|PREFIX

FIELDWORD

Details about the word closest to the cursor in the current cursor field.
(E)

fieldword.0 − 2

fieldword.1 − word as defined by SET WORD ALPHANUM

fieldword.2 − word as defined by SET WORD NONBLANK

FILEName

The full filename of the current file, including any file extension.
(QEM)

filename.0 − 1

filename.1 − Full file name.

FILESTATUS

Details about the status of the file being edited.
(QEM)

filestatus.0 − 3

filestatus.1 − sharing mode − NONE

filestatus.2 − access type − READONLY|READWRITE

filestatus.3 − end of line − CR/LF/CRLF/NONE

FMode

The file mode of the current file. Under Un*x, this will always return an empty string. Other platforms returns the first character of the file *s path. ie the disk drive.*
(QEM)

fmode.0 − 1

fmode.1 − File mode.

FName

The fname portion of the current file. See SET FILENAME for a full description of the components of a file name.
(QEM)

fname.0 − 1

fname.1 − File name.

FPath

The path name of the current file. This includes a trailing directory separator.
(QEM)

fpath.0 − 1

fpath.1 − File path.

FType

The extension of the current file. The characters following the trailing . character.
(QEM)

ftype.0 − 1

ftype.1 − File extension.

FULLFName

Indicates if the fully qualified filename is displayed on the idline .
(QEMS)

fullfname.0 − 1

fullfname.1 − ON|OFF

GETENV variable

The value of the supplied environment *variable* or ***invalid*** if the *variable* does not exist. On platforms other than Unix the supplied variable name is uppercased before obtaining the environment variable value.
(E)

getenv.0 − 1

getenv.1 − value of variable

HEX

Indicates if hexadecimal values in commands are interpreted as hexadecimal values or not. Set by SET HEX .
(QEMS)

hex.0 − 1

hex.1 − ON|OFF

HEADer

Returns details about which sections of a TLD file are to be applied to the current view. Do not rely on the position of a particular header in the returned stem. Set by SET HEADER .
(QE)

header.0	−	11
header.1	−	section ON|OFF
header.2	−	section ON|OFF
header.3	−	section ON|OFF
header.4	−	section ON|OFF
header.5	−	section ON|OFF
header.6	−	section ON|OFF
header.7	−	section ON|OFF
header.8	−	section ON|OFF
header.9	−	section ON|OFF
header.10	−	section ON|OFF
header.11	−	section ON|OFF

HEXDISPlay

Indicates if the current character is displayed on the status line . Set by SET HEXDISPLAY .
(QEMS)

hexdisplay.0	−	1
hexdisplay.1	−	ON|OFF

HEXShow

Returns details of HEXSHOW placement. Set by SET HEXSHOW .
(QEMS)

hexshow.0	−	2
hexshow.1	−	ON|OFF
hexshow.2	−	line displaying first hexshow line

HIGHlight

Returns details of HIGHLIGHT settings. Set by SET HIGHLIGHT .
(QEMS)

highlight.0	−	1 or 3 (if highlight.1 = SELECT)
highlight.1	−	OFF|ALTERED|TAGGED|SELECT
highlight.2	−	minimum (or only) selection level for SELECT
highlight.3	−	maximum selection level for SELECT

IDline

Indicates if the idline is displayed for a file. Set by SET IDLINE .
(QEMS)

idline.0	−	1
idline.1	−	ON|OFF

IMPMACro

Indicates if implied macro processing is on or off. Set by SET IMPMACRO .
(QEMS)

impmacro.0	−	1
impmacro.1	−	ON|OFF

IMPOS

Indicates if implied operating system command processing is on or off. Set by SET IMPOS or SET IMPCMSCP .
(QEMS)

impos.0 − 1

impos.1 − ON|OFF

INPUTMode

Indicates the inputmode for the current view. Set by SET INPUTMODE .
(QEMS)

inputmode.0 − 1

inputmode.1 − OFF|FULL|LINE

INSERTmode

Indicates if currently in insert mode or overstrike mode. Set by SET INSERTMODE .
(QEMS)

insertmode.1 − 1

insertmode.1 − ON|OFF

LASTmsg

Return the text of the last error message generated.
(E)

lastmsg.0 − 1

lastmsg.1 − text of last message.

LASTKEY [n]

Return information about the last key, or the nth last key pressed.
(E)

lastkey.0 − 4

lastkey.1 − name of the key.

lastkey.2 − character associated with the key (if applicable)

lastkey.3 − curses mnemonic code

lastkey.4 − shift status. see READV for details

LASTOP [command|*]

Returns the last operand from selected commands. command can be one of ALter, Change, CLocate COUnt, Find, Locate SCHange or TFind.
(QE)
EXTract /LASTOP command/ sets:

lastop.0 − 1

lastop.1 − command and its last operand

If no command is supplied as a parameter or * is passed, details of all last operands are set as follows:

lastop.0 − 8

lastop.1 − "alter" and its last operand (not implemented)

lastop.2 − "change" and its last operand

lastop.3 − "clocate" and its last operand (not implemented)

lastop.4 − "count" and its last operand (not implemented)

lastop.5 − "find" and its last operand

lastop.6 − "locate" and its last operand

lastop.7 − "schange" and its last operand

lastop.8 − "tfind" and its last operand (not implemented)

LASTRC

Returns the return code from last command issued from command line .
(QES)

lastrc.0 − 1

lastrc.1 − Last return code.

LENgth

Length of the current line.
(QES)

length.0 − 1

length.1 − Length of current line.

LIne

Line number of focus line in current file; or current line if SET COMPAT (feel) is set to XEDIT.
(QES)

line.0 − 1

line.1 − Line number

LINEFLAG

Returns information about the flags set on the focus line , or current line if SET COMPAT (feel) is set to XEDIT.
(QEMS)

lineflag.0 − 3

lineflag.1 − NEW|NONEW

lineflag.2 − CHANGE|NOCHANGE

lineflag.3 − TAG|NOTAG

LINENd

Indicates if multiple commands allowed on command line and the delimiter. Set by SET LINEND .
(QEMS)

linend.0 − 2

linend.1 − ON|OFF

linend.2 − delimiter

LScreen

Displays the size of the current screen and window. Also shows the upper left corner of the window.
(QE)

lscreen.0 − 6

lscreen.1 − height of current screen

lscreen.2 − width of current screen

lscreen.3 − screen line of upper left corner of screen

lscreen.4 − screen column of upper left corner of screen

lscreen.5 − height of display

lscreen.6 − width of display

MACRO

Indicates if macros are executed before commands. Set by SET MACRO .
(QEMS)

macro.0 − 1

macro.1 − ON|OFF

MACROExt

The current setting for a macro *s file extension. Set by SET MACROEXT* .
(QEMS)

macroext.0 − 1

macroext.1 − Default file extension

MACROPath

The path that THE looks for by default for macro files. Set by SET MACROPATH .
(QEMS)

macropath.0 − 1

macropath.1 − Path for macro files.

MARgins

The settings for left and right margins and paragraph indent. Set by SET MARGINS .
(QEMS)

margins.0 − 3

margins.1 − left column

margins.2 − right column

margins.3 − indent value (column or offset from left margin)

MONITOR

Indicates if the combination of monitor and the curses package supports colour. If the curses package supports colour, then monitor.1 is set to COLOR and monitor.2 can be COLOR or MONO depending on whether the monitor supports colour. If monitor.1 is MONO then monitor.2 will also be set to MONO.
(QE)

monitor.0 − 2

monitor.1 − COLOR|MONO

monitor.2 − COLOR|MONO

MOUSE

Indicates if the mouse is suppoted as an input device. Set by SET MOUSE .
(QEMS)

mouse.0 − 1

mouse.1 − ON|OFF

MSGLine

Returns details of where the message line is displayed. Set by SET MSGLINE .
(QEMS)

msgline.0 − 4

msgline.1 − ON

msgline.2 − line position of message line

msgline.3 − number of message lines available

msgline.4 − OVERLAY (returned for compatibility reasons)

MSGMode

Indicates if messages are suppressed. Set by SET MSGMODE .
(QEMS)

msgmode.0 − 1

msgmode.1 − ON|OFF

NBFile

Returns with the number of files currently in the ring .

(QES)

nbfile.0 − 1

nbfile.1 − Number of files in ring

NBScope

Returns with the number of lines currently in scope. If SCOPE ALL is in effect, this will be the same number as SIZE.

(QE)

nbscope.0 − 1

nbscope.1 − Number of lines in scope

nbscope.2 − Line number of focus line in scope

NEWLines

Indicates if NEWLINES variable is set to LEFT or ALIGNED. Set by SET NEWLINES .

(QEMS)

newlines.0 − 1

newlines.1 − ALIGNED|LEFT

NONDisp

Returns the character that is displayed for extended characters that are not displayed. Set by SET NONDISP .

(QEMS)

nondisp.0 − 1

nondisp.1 − char

NUMber

Indicates if line numbers are displayed in the prefix area. Set by SET NUMBER .

(QEMS)

number.0 − 1

number.1 − ON|OFF

PAGEWRAP

Indicates if the scrolling the file view using the FORWARD and BACKWARD commands will wrap if the cursor is at the Bottom−of−File line or Top−of−File line respectively. Set by SET PAGEWRAP .

(QEMS)

pagewrap.0 − 1

pagewrap.1 − ON|OFF

PARSER [*|parser]

Displays the filename of the THE Language Defintion file from which a syntax highlighting parser was loaded. Set by SET PARSER .

(QES)

parser.0 − 2

parser.1 − Name of parser

parser.2 − Filename of TLD file

If no parser is supplied as a parameter or * is passed, details of all parsers are set as follows:

parser.0 − number of parsers currently defined

parser.1 − name and filename of first parser

parser.i − name and filename of ith parser

PENDing [BLOCK] [OLDNAME] name|* [target1 [target2]

Returns information about pending prefix commands.

(E)

pending.0	−	4
pending.1	−	line number in file
pending.2	−	newname − actual name entered in prefix area
pending.3	−	oldname − original name of macro after synonym resolution
pending.4	−	BLOCK or null

Point [*]

Returns the name and line number of the focus line , or names and line numbers of all lines in a file if * is specified. If SET COMPAT (feel) is set to XEDIT, then the name and line number of the current line is returned, rather than the name and line number of the focus line .

(QE)

With no arguments:

point.0	−	0 or 1 (0 if focus line not named)
point.1	−	line number and name of line (if line is named)

With [*] argument:

(E)

point.0	−	number of named lines in the file
point.1	−	line number and name for first named line
point.n	−	line number and name for nth named line

Only point.0 and point.1 are available using implied extract functions.

POSition

Indicates if LINE/COL is displayed on idline . Set by SET POSITION .

(QMS)

position.0	−	1
position.1	−	ON|OFF

The current/focus line/column is also returned via EXTRACT .

(E)

position.0	−	3
position.1	−	ON|OFF
position.2	−	current or focus line
position.3	−	current or focus column

PREfix [Synonym [*]]

Indicates if prefix is displayed for the view and if displayed where is is displayed. See SET PREFIX .

(QEMS)

prefix.0	−	1 or 2 (1 if prefix.1 is OFF, 2 otherwise)
prefix.1	−	ON|OFF|NULLS
prefix.2	−	LEFT|RIGHT (if prefix.1 is ON or NULLS)
prefix.3	−	width of prefix area
prefix.4	−	width of prefix gap

With [*Synonym*] option, the name of the macrofile (oldname) is returned that is associated with the synonym. If name is not a synonym then name is returned.

(E)

prefix.0	−	1
prefix.1	−	oldname

With [*Synonym* [*]] option, all prefix synonyms are returned.

(E)

prefix.0 − number of prefix synonyms

prefix.1 − newname oldname

prefix.n − newname oldname

PRINTER

Returns the value of the printer port or spooler. Set by SET PRINTER

(QEMS)

printer.0 − 1

printer.1 − port or spooler name

READONLY

Indicates if the file being edited is allowed to be alterered. Set by SET READONLY .

(QEMS)

readonly.0 − 1

readonly.1 − ON|OFF|FORCE

REGEXP

Return the type of regular expression syntax currently used for targets. Set by SET REGEXP .

(QEMS)

regexp.0 − 1

regexp.1 − EMACS|AWK|POSIX_AWK|GREP|EGREP|POSIX_EGREP|SED|POSIX_BASIC

− |POSIX_MINIMAL_BASIC|POSIX_EXTENDED|POSIX_MINIMAL_EXTENDED

REPROFile

Indicates if the specified (or implied) profile file is re−executed each time a file is added to the ring. Set by SET REPROFILE .

(QEMS)

reprofle.0 − 1

reprofle.1 − ON|OFF

RESERved [*]

Return with list of the screen rows that have been reserved. Set by SET RESERVED .

(QES)

reserved.0 − 0 if no reserved lines; 1 otherwise

reserved.1 − list of screen row numbers reserved

With [*] option, the line number, followed by the colour specification and reserved line contents are returned; one variable for each reserved line.

(E)

reserved.0 − the number of reserved lines

reserved.1 − first reserved line

reserved.2 − second reserved line

reserved.n − nth reserved line

REXX

Returns the version details of the Rexx interpreter (if any) in the same format as PARSE VERSION.

(QE)

rexx.0 − 1

rexx.1 − Version strings

REXXOUTput

Indicates if REXX output is captured to a file or not and the line number limit of lines to be displayed. Set by SET REXXOUTPUT .

(QEMS)

rexxoutput.0 − 2

rexxoutput.1 − FILE|DISPLAY

rexxoutput.2 − line number limit

RING

Returns details of each file being edited.
(QE)
With SET COMPAT (feel) set to XEDIT, the values set are:

ring.0 − number of variables returned (ring.1 + 1)

ring.1 − number of files in the ring

ring.2 − IDLINE of first file in the ring

ring.3 − IDLINE of second file in the ring

ring.n − IDLINE of nth file in the ring

With SET COMPAT (feel) set to THE or KEDIT, the values set are:

ring.0 − number of files in the ring

ring.1 − IDLINE of first file in the ring

ring.2 − IDLINE of second file in the ring

ring.n − IDLINE of nth file in the ring

Only ring.0 and ring.1 are available using implied extract functions.

SCALe

Returns details of scale line . Set by SET SCALE .
(QEMS)

scale.0 − 2

scale.1 − ON|OFF

scale.2 − line displaying scale line

SCOPE

Returns information about whether shadow line s will be affected by commands or not. Set by SET SCOPE .
(QEMS)

scope.0 − 1

scope.1 − ALL|DISPLAY

SCReen

Returns the number and orientation of THE screens. Set by SET SCREEN .
(QEMS)

screen.0 − 2

screen.1 − Number of screens displayed

screen.2 − HORIZONTAL|VERTICAL

SELect

Returns the selection level of the focus line and the maximum selection level for the file. Set by SET SELECT and ALL commands. If SET COMPAT (feel) XEDIT is set, then the selection level of the current line is returned instead of the selection level of the focus line .
(QES)

select.0 − 2

select.1 − selection level of focus line

select.2 − maximum selection level for file

SHADow

Returns the status of the display of shadow lines. Set by SET SHADOW .
(QEMS)

shadow.0 − 1

shadow.1 − ON|OFF

SHOWkey key

Returns the commands and parameters assigned to the *key* passed as an argument. This keyvalue is returned as READV.3 from a call to READV KEY.

(E)

showkey.0 − the number of commands assigned

showkey.1 − first command/parameter assigned

showkey.n − last command/parameter assigned

If no *key* supplied as a parameter:

showkey.0 − 1

showkey.1 − INVALID KEY

This item is not available as an implied extract function.

SIze

Returns the number of lines in the current file.

(QES)

size.0 − 1

size.1 − Lines in current file.

STATUSLine

Indicates if the status line is displayed and if so, where. Set by SET STATUSLINE .

(QEMS)

statusline.0 − 1

statusline.1 − TOP|BOTTOM|OFF

STAY

Indicates if the focus line stays where it is after a successful THE command or an unsuccessful LOCATE command. Set by SET STAY .

(QEMS)

stay.0 − 1

stay.1 − ON|OFF

SYNonym [*|name]

Indicates if synonym processing is on or off. Set by SET SYNONYM command.

(QEMS)

synonym.0 − 1

synonym.1 − ON|OFF

With [*name*] option, details about the defined synonym are returned.

(QE)

synonym.0 − 4

synonym.1 − synonym name

synonym.2 − length of minimum abbreviation

synonym.3 − definition

synonym.4 − linend character (if specified)

With [*] option, details about all defined synonyms are returned.

(QE)

synonym.0 − number of synonyms defined

synonym.1 − synonym definition of first synonym

synonym.n − synonym definition of nth synonym

TABKey

Returns settings about behaviour of TAB key. tabkey.1 indicates behaviour while not in insert mode tabkey.2 indicates behaviour while in insert mode Set by SET TABKEY .
(QEMS)

tabkey.0 − 2

tabkey.1 − TAB|CHARACTER

tabkey.2 − TAB|CHARACTER

TABLine

Returns details of if and where the tab line is displayed. Set by SET TABLINE .
(QEMS)

tabline.0 − 2

tabline.1 − ON|OFF

tabline.2 − line displaying tab line

TABS

Returns settings about tab columns. Set by SET TABS .
(QEMS)

tabs.0 − 1

tabs.1 − actual tab columns or "INCR n"

TABSIn

Indicates if TABSIN processing is on or off and the size of the tabs setting. Set by SET TABSIN .
(QEMS)

tabsin.0 − 2

tabsin.1 − ON|OFF

tabsin.2 − size of tabs

TABSOut

Indicates if TABSOUT processing is on or off and the size of the tabs setting. Set by SET TABSOUT .
(QEMS)

tabsout.0 − 2

tabsout.1 − ON|OFF

tabsout.2 − size of tabs

TARGETSAVE

Indicates the type of targets saved for subsequent LOCATE commands. Set by SET TARGETSAVE .
(QEMS)

targetsave.0 − 1

targetsave.1 − ALL|NONE|list of target types

TERMinal

Identifies the terminal type currently being used.
(QES)

terminal.0 − 1

terminal.1 − DOS|OS2|X11|WIN32|$TERM value under Unix

THIGHlight

Indicates if text highlighting is on. ie a found string target is highlighted. Set by SET THIGHLIGHT
(QEMS)

thighlight.0 − 1

thighlight.1 − ON|OFF

TIMECHECK

Indicates the status of TIMECHECK for the current file.
(QEMS)

timecheck.0 − 1

timecheck.1 − ON|OFF

TOF

Indicates if the current line is on the Top−of−File line .
(QES)

tof.0 − 1

tof.1 − ON|OFF

TOFEOF

Indicates if the Top−of−File line and the Bottom−of−File line are displayed.
(QEMS)

tofeof.0 − 1

tofeof.1 − ON|OFF

TRAILING

Indicates how trailing blanks on lines are handled when the file is saved. Set by SET TRAILING
(QEMS)

trailing.0 − 1

trailing.1 − ON|OFF|EMPTY|SINGLE

TYPEAhead

Indicates if THE will wait until all keyboard input has been processed before updating the screen display. Set by SET TYPEAHEAD .
(QEMS)

typeahead.0 − 1

typeahead.1 − ON|OFF

UNDOING

Indicates if the undo facility is available or not. Set by SET UNDOING
(QEMS)

undoing.0 − 1

undoing.1 − ON|OFF

UNTAA

Indicates if "unsigned numbers are absolute". Set by SET UNTAA
(QEMS)

untaa.0 − 1

untaa.1 − ON|OFF

Verify

Returns verify column settings. Set by SET VERIFY .
(QEMS)

verify.0 − 1

verify.1 − Column pair of verify start and end columns.

VERShift

Returns the value of the vershift internal variable.
(QES)

vershift.0 − 1

vershift.1 − VERSHIFT value

VERSION

Returns information about name of application (THE) and version information.
(QES)

version.0 − 4

version.1 − THE

version.2 − version string eg. 1.5

version.3 − platform version (DOS,OS2,UNIX,X11,WIN32)

version.4 − version status information eg. release date, beta

Width

Returns maximum line width setting. Set by −w command line switch on starting THE.
(QEMS)

width.0 − 1

width.1 − Maximum line width value.

WORD

Specifies how THE defines a word. Set by SET WORD .
(QEMS)

word.0 − 1

word.1 − ALPHANUM|NONBLANK

WORDWrap

Indicates if WORDWRAP is on or off. Set by SET WORDWRAP .
(QEMS)

wordwrap.0 − 1

wordwrap.1 − ON|OFF

WRap

Indicates if WRAP is on or off. Set by SET WRAP .
(QEMS)

wrap.0 − 1

wrap.1 − ON|OFF

XTERMinal

Returns the current value of the X11 terminal program. Only applicable in X version. Set by SET XTERMINAL .
(QEMS)

xterminal.0 − 1

xterminal.1 − X11 terminal program

Zone

Returns zone column settings. Set by SET ZONE .
(QEMS)

zone.0 − 2

zone.1 − Zone start column

zone.2 − Zone end column

IMPLIED EXTRACT

The above REXX variables set by the EXTRACT command may also be obtained by a REXX macro as an implied EXTRACT. Each variable above that may be set by an explicit EXTRACT command may also be eg. The REXX commands:

```
        'EXTRACT /SIZE/CURLINE/'
        Say size.1 curline.1
```

may be substituted with:

```
        Say size.1() curline.1()
```

BOOLEAN FUNCTIONS

THE also provides other information to the REXX interpreter via boolean functions. These functions return either *1* or *0* depending on the information queried.

after()

Returns *1* if the cursor is currently after the last non−blank character on the line, or if the line is blank.

altkey()

Returns *1* if at the time the last key was pressed, the ALT key was also being held down.

alt()

Returns *1* if the file being edited has changed since the last SAVE. ie. if the value of alt.2 is non zero.

blank()

Returns *1* if the line the cursor field is completely blank.

batch()

Returns *1* if THE is being run in batch mode. ie THE was started with the −b switch.

block()

Returns *1* if the marked block is within the current view.

before()

Returns *1* if the cursor is currently before the first non−blank character on the line, or if the line is blank.

bottomedge()

Returns *1* if the cursor is on the bottom edge of the filearea or prefix area .

command()

Returns *1* if the command line is on. This is different to the definition in KEDIT; "Returns *1* if the cursor is on the command line." To get the equivalent KEDIT functionality use incommand();

ctrl()

Returns *1* if at the time the last key was pressed, the CTRL key was also being held down.

current()

Returns *1* if the cursor is on the current line .

dir()

Returns *1* if the current file is the special DIR.DIR file.

end()

Returns *1* if the cursor is on the last non−blank character on the line.

eof()

Returns *1* if the cursor is on the Bottom−of−File line and the cursor is not on the command line .

first()

Returns *1* if the cursor is in column 1 of the current window.

focuseof()

Returns *1* if the focus line is the Bottom−of−File line whether the cursor is on it or not.

focustof()

Returns *1* if the focus line is the Top−of−File line whether the cursor is on it or not.

inblock()

Returns *1* if the cursor is in the marked block .

incommand()

Returns *1* if the cursor is on the command line .

initial()

Returns *1* if the function is called from the profile .

inprefix()

Returns *1* if the cursor is located in the prefix area .

leftedge()

Returns *1* if the cursor is on the left edge of the filearea .

modifiable()

Returns *1* if the cursor is located in an area that can be changed. ie. not on Top−of−File line or Bottom−of−File line nor on a shadow line .

rightedge()

Returns *1* if the cursor is on the right edge of the filearea .

shadow()

Returns *1* if the cursor is on a shadow line .

shift()

Returns *1* if at the time the last key was pressed, the SHIFT key was also being held down.

spacechar()

Returns *1* if the cursor is on a space character.

tof()

Returns *1* if the cursor is on the Top−of−File line and the cursor is not on the command line .

topedge()

Returns *1* if the cursor is on the top edge of the filearea .

verone()

Returns *1* if the column 1 of the file is being displayed in column 1.

OTHER FUNCTIONS

The following functions provide features to simplify THE macros written in REXX.

valid_target(target[,anything])

The first argument is the target to be validated. If a second, optional argument is supplied, the target to be validated can consist of a target followed by any optional characters. This can be useful if the arguments to a macro consist of a target followed by another argument. If a valid target is supplied, the remainder of the string passed to valid_target() is returned following the first line affected and the number of lines to the target.

Returns *ERROR* if the supplied target is invalid. Returns *NOTFOUND* if the supplied target is valid, but not found.

If a valid target, returns the first line affected by the target followed by the number of lines to the target, and optionally the remainder of the argument. eg.

if the focus line is 12 and valid_target() is called as

```
result = valid_target(":7") ===> result = "12 −5"
```

if the focus line is 12 and valid_target is called as

```
result = valid_target(":7 /fred/",junk), ===> result = "12 −5 /fred/"
```

run_os(command[,stdin_stem[,stdout_stem[,stderr_stem]]])

This function allows the macro writer to call an operating system command and have the standard streams; *stdin* , *stdout* and *stderr* redirected to or from REXX arrays.
The first argument is the operating system command to execute. The command can include any command line switches appropriate to the command.
All other arguments comprise a stem name (including a trailing .) which refers to the REXX arrays where *stdin* , *stdout* and *stderr* are to be redirected.
As with all REXX arrays, the value of the 0th element (stem.0) contains the number of elements in the array.
The only restriction with the names of the stem variables is that the stem name for the *stdin* stem cannot be the same as the stem for *stdout* or *stderr* .
The stem name for *stdout* and *stderr* can be the same; the contents of the resulting output stems will consist of *stdout* and *stderr* in the order that the command generates this output.
Return values:
0 − successful
1005 − invalid argument or syntax
1012 − problems with system redirection of streams
1094 − out of memory
1099 − error interfacing to REXX interpreter
all other numbers, return code from operating system command
eg. to spell check the words "The Hessling Editr" with ispell

```
in.0 = 3
in.1 = "The"
in.2 = "Hessling"
in.3 = "Editr"
rc = run_os("ispell −a","in.","out.")
```

sets:
out.0 −−> 3
out.1 −−> "*"
out.2 −−> "#"
out.3 −−> "&edits edit editor"

Return to Table of Contents

GLOSSARY

block

A portion of the file being edited; usually highlighted. A block can be one of line block , box block , word block , column block , or stream block .

Bottom−of−File line

A line marker, identifying the end of the file.

box block

A type of block . It is a rectangular portion of the filearea .

column block

A type of block . It is similar to a box block except that a columnar portion of every line in the file is included.

column target

A method of referring to some part of a line being edited.

command line

The area of the display, usually identified by an arrow prompt, "====>", where THE commands are entered.

current column

The column in the filearea , marked by the "|" character on the scale line .

current line

The line in the filearea , normally highlighted, from which commands entered on the command line take effect.

cursor field

The "field" in which the cursor is currently located. This can be the prefix area , command line or filearea .

filearea

That part of the screen display where the contents of the file being edited are displayed.

focus column

The column in the filearea which currently has the focus. If the cursor is in the filearea , it is the column on which the cursor is displayed. If the cursor is in the command line or prefix area , the focus column is the current column .

focus line

The line in the filearea which currently has the focus. If the cursor is in the filearea or prefix area , it is the line on which the cursor is displayed. If the cursor is on the command line , the focus line is the current line . All THE commands operate relative to the focus line.

idline

That part of the screen display which shows details about the file being edited.

line block

A type of block . It consists of consecutive lines within the filearea .

macro

A file containing a sequence of THE commands. Macros can be written in REXX if a supported REXX interpreter is available.
*** NOTE ***
If REXX support is NOT enabled in THE, the first line of the THE macro file MUST contain the comment:
/*NOREXX*/

magic number

A magic number is used on Un*x systems to identify the type of a file. It was originally used in binary files to identify the file, but the concept was also extended to text files; particularly shell scripts; to enable to shell to determine which interpreter should be used to execute the contents of a file. These magic numbers in text files are really magic "strings" and it is these strings that are used by the THE syntax highlighting feature to automatically determine the parser to be used to highlight a file.

message line

That portion of the screen used to display messages.

prefix area

That part of the screen display where prefix commands can be entered. It also displays the line number of each line in the file if SET NUMBER is ON.

parser

A mechanism that translates text strings into identifiable tokens.

profile

A macro file that is executed at the start of an editing session.

relative target

A subset of target , but only supports targets that are relative to the focus line , like 3, −5, *.

reserved line

A line within the filearea that contains user text that the user can define with the SET RESERVED command.

REXX

A powerful, easy−to−learn macro language available on most platforms. THE uses REXX as its macro language.

ring

The set of files currently being edited are arranged in a ring formation.

scale line

A line in the filearea showing column positions.

shadow line

A line which signifies how many lines have been excluded from display. See SET SHADOW , SET SELECT .

status line

That part of the screen display showing details about the entire THE session.

stream block

A type of block . It is a stream of characters that span one or more consecutive lines in the filearea .

string target

A subset of target that refers to a string of characters enclosed between string target delimiter s.

string target delimiter

The character that determines the start and optional ending of a string target . A delimiter can consist of one of the following characters: /\@`#$%(){}[]"\

tab line

A line in the filearea which displays the currently set TAB stops.

target

A method of referring to some part of the file being edited.

Top−of−File line

A line marker, identifying the beginning of the file.

vershift

An internal number which defines the number of columns the displayed text is offset from the verify setting as a result of automatically scrolling horizontally, or by the use of the LEFT , RIGHT , or RGTLEFT commands.

word block

A type of block . It consists of a contiguous series of characters that comprise a word. A word is defined by the current setting of SET WORD .

Generated on: 2 Nov 2002

Return to Table of Contents

APPENDIX 1 − ENVIRONMENT VARIABLES

THE uses the following environment variables:

THE_HOME_DIR
the directory in which THE looks for its help file(s) and macro(s)

THE_HELP_FILE
the fully qualified file name containing help information.

THE_PROFILE_FILE
the fully qualified file name for the default profile file.

THE_MACRO_PATH
a list of all directories in which THE is to look for macro files.

Environment Variable Use under UNIX

If no environment variable; THE_HOME_DIR, is set prior to starting THE, THE uses /usr/local/THE/ as the default value for ${THE_HOME_DIR} if running under Unix. This can be changed when running the configure script. See the file INSTALL for further details.

The default value of THE_HELP_FILE is THE_Help.txt. With no environment variables set, the help file to be looked for will be /usr/local/THE/THE_Help.txt. If THE_HOME_DIR environment variable was set to /usr/opt/etc or changed in the.c, the help file will be /usr/opt/etc/THE_Help.txt.

The name of the default profile file is $HOME/.therc.

The default value of THE_MACRO_PATH is ${THE_HOME_DIR}:. THE will look for a macro file first in ${THE_HOME_DIR} and then in the current directory. So with no environment variables set, THE looks for macros in $(THE_HOME_DIR} first and if no file exists in that directory, THE will look in the current directory. If the file does not exist in either of these places, an error message will be displayed. The format of THE_MACRO_PATH is dir1:dir2:dir3 Each directory is seperated by colons. By default, up to 20 directories may be specified. This can be changed in the.h when building or by the SET MACROPATH command.

Environment Variable Use under OS/2, DOS and Win95/NT

****

Under some circumstances the following default behaviour does not occur. The reason is that THE.EXE sometimes cannot determine where it was run from. This is a due to the different ways that different operating systems behave and how the C compiler used to build THE behaves.

It is probably better to explicitly set the environment variables than hope that your particular environment will work as documented below.

****

By default THE_HOME_DIR is set to the directory from which THE was invoked. Assuming THE.EXE resides in d:\tools, THE_HOME_DIR will equate to d:\tools\.

The default value of THE_HELP_FILE is THE_Help.txt.

The name of the default profile file is ${THE_HOME_DIR}profile.the.

The default value of THE_MACRO_PATH is ${THE_HOME_DIR};. THE will look for a macro file first in THE_HOME_DIR and then in the current directory. So with no environment variables set, THE first looks for macros in the directory from which THE was invoked and then in the current directory. If the file does not exist in either directory, an error message is displayed. The format of THE_MACRO_PATH is dir1;dir2;dir3 Each directory is seperated by semi−colons. By default, up to 20 directories may be specified. This can be changed by the SET MACROPATH command.

Generated on: 2 Nov 2002

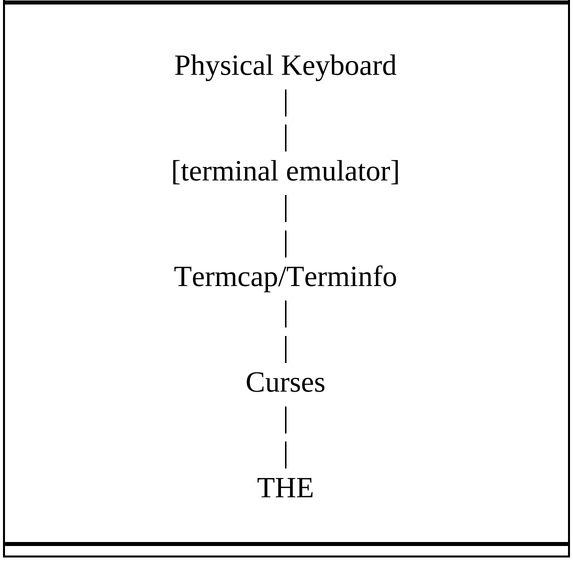
Return to Table of Contents

APPENDIX 2 − KEYBOARD HANDLING IN THE

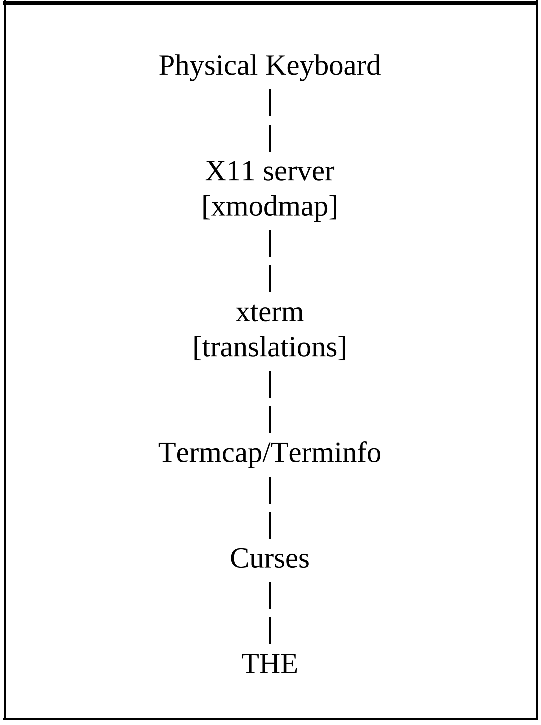
This appendix contains information on how THE handles keystrokes in the U*ix environment. One thing that is consistant with PCs is keyboard handling. Therefore this explanation is not applicable to the PC arena.

Keystroke handling is a very complicated business! There are so many layers between the physical keyboard and the application; in this case THE. These layers can be best described with the following diagrams:

Text−mode Version of THE using terminal



Text−mode Version of THE using xterm



X11 Version of THE
Physical Keyboard | | X11 server [xmodmap] | | PDCurses [translations] | | THE

Each layer is described below

Physical keyboard

Each keyboard is potentially different from every other. The similarity that they do posess is that when a key is pressed, a unique code is generated. The state of the Shift, Control, Num Lock, Alt modifiers either alter the unique code generated or a seperate code is generated to identify to the next layer that the key pressed should be modified.

terminal emulator

This is a required layer if the user is connected to the host via some emulation software. eg a PC connected to a U*ix host requires a terminal emulator of one sort or another.

The terminal emulator translates the unique codes generated by the keyboard usually into physical escape sequences that are passed to the U*ix shell. These escape sequences can be seen by running the U*ix command *cat −v* . When you press a key on the keyboard while in *cat −v* , the escape sequences that the terminal emulator has generated are displayed. Some systems require the Enter key to be pressed before the escape sequence is displayed. The term "escape sequence" is used because in most cases, the first character generated is an Escape character (hex 1B). Following the escape is any number of other characters, which together form a unique sequence of characters.

When using a terminal directly, firmware on the terminal does the generation of the escape sequences. Some terminals allow the user to specify what escape sequence will be generated when particular keys are pressed.

Termcap/Terminfo

Each terminal that connects to a U*ix host is identified by the value of the TERM environment variable. The value of TERM is used to setup various settings for the terminal. These settings are stored in a database; either in a flat file *termcap* or in a compiled format *terminfo* . Which one a particular system uses is dependant on the version of the U*ix Operating System being run.

One of the capabilites of the termcap/terminfo databases is to translate an escape sequence into a keyboard mnemonic that is used by the Curses library functions. This translation can also be changed by the user.

An example of this is:

```
Termcap: ku=\E[A
Terminfo: kcuu1=\E[A
```

This translates the escape sequence ESC [A into the mnemonic ku or kccu1.

Curses

The Curses library contains definitions of many keys. These can be seen by looking in the Curses header file; usually /usr/include/curses.h Here you will see defintions like:

```
#define KEY_UP 0403 /* up arrow key */
```

THE

As THE is a Curses−based application, it recognises the Curses key definitions defined in curses.h. As not all escape sequences are usually defined in the Termcap/Terminfo databases, or are defined in curses.h, THE also has special code to decipher other escape sequences.

When a key is pressed, and is interpreted by THE, the first action is to check if the value passed to THE is a *known* curses key. If it is THE carries out any command associated with the key. If the value is not a *known* curses key, THE assumes that an escape sequence is forthcoming. The module getch.c has code for deciphering many escape sequences that are not normally defined in the Termcap/Terminfo database. Later in this document is a table of escape sequences and its associated curses key.

X11 Server

When THE is run in a X environemnt, the X server acts in a similar way to terminal emulation software. The principal difference is that there are more layers within the X server. The translation of physical keyboard codes to an X mnemonic is the first layer. The code generated is called a keycode. This keycode is then translated into another higher level mnemonic; a keysym. The keysym is usually a name that resembles the label on the physical keyboard.

The X environment provides a mechanism to assign keycodes to keysyms. This is done with the xmodmap command, and any assignment of keycode to keysym is done for all applications within the current X environemnt.

xterm

The xterm program is a terminal emulation program, and as such has many of the properties of the terminal emulator described above. The xterm program takes keysyms from the X11 server and generates escape sequences, which are defined in the xterm entry in the Termcap/ Terminfo database.

One of the features of xterm is the ability to alter the standard keysym to escape sequence translation. Surprisingly this is via a feature called *translations* . Not only do these translations enable the user to specify a different escape sequences to be generated, but internal xterm commands can also be assigned. These are beyond the scope of this description. An example of an xterm translation follows:

```
*VT100.Translations: #override\n\
<Key>F1: string(0x1b) string("[192z")\n\
<Key>F2: string(0x1b) string("[193z")\n
```

This specification is typically part of your own $HOME/.Xdefaults file. This file is usually read dynamically by the xterm program on startup. On some systems, the .Xdefaults file is ignored. To ensure the entries are incorporated into the X server resource database, run the command:

```
xrdb −merge .Xdefaults
```

The above example will generate the escape sequence ESC [1 9 2 z when the F1 key is pressed. The trailing "\" is a continuation character.

So, if you had the above translation in effect, and were running THE in an xterm, and ran the SHOWKEY command, pressing the F1 key would result in THE responding with F11.

PDCurses

The actions performed by PDCurses in the X environment combine the xterm, Termcap/Terminfo and curses actions. This results in fewer layers to be traversed.

PDCurses takes a keysym from the X server and converts it into a Curses key code. PDCurses also has the same translation capabilities as does xterm. The xterm example above would look like:

```
*the.Translations: #override\n\
<Key>F1: string(0x1b) string("[192z")\n\
<Key>F2: string(0x1b) string("[193z")\n
```

All the details described in xterm as far as translations are concerned hold for translations in PDCurses.

Another capability that translations offer is the ability to assign international characters to alphabetic characters, while still having the alphabetic characters available. For example, the translations below assigns the action of the Shifted Meta key and A, giving a capital A with a grave accent, and the unshifted Meta key and A giving a small A with a grave accent.

```
*the.Translations: #override\n\
!Shift Meta <Key>a: string(0xc0) \n\
!Meta <Key>a: string(0xe0) \n
```

The examples above assume you are using a font with a character set that is compatible with ISO 8859−1.

THE Escape Sequence to Key Name Mapping

Escape Sequence	Curses Key	THE Key Name
(pre) 1 ~	KEY_Find	FIND
(pre) 2 ~	KEY_InsertHere	INS
(pre) 3 ~	KEY_Remove	DEL
(pre) 4 ~	KEY_Select	SELECT
(pre) 5 ~	KEY_PrevScreen	PGUP
(pre) 6 ~	KEY_NextScreen	PGDN
(pre) 1 1 ~	KEY_F(1)	F1
(pre) 1 2 ~	KEY_F(2)	F2
(pre) 1 3 ~	KEY_F(3)	F3
(pre) 1 4 ~	KEY_F(4)	F4
(pre) 1 5 ~	KEY_F(5)	F5
(pre) 1 7 ~	KEY_F(6)	F6
(pre) 1 8 ~	KEY_F(7)	F7
(pre) 1 9 ~	KEY_F(8)	F8

(pre) 2 0 ~	KEY_F(9)	F9
(pre) 2 1 ~	KEY_F(10)	F10
(pre) 2 3 ~	KEY_F(11)	F11
(pre) 2 4 ~	KEY_F(12)	F12
(pre) 2 5 ~	KEY_F(49)	F13
(pre) 2 6 ~	KEY_F(50)	F14
(pre) 2 8 ~	KEY_F(51)	F15
(pre) 2 9 ~	KEY_F(52)	F16
(pre) 3 1 ~	KEY_F(53)	F17
(pre) 3 2 ~	KEY_F(54)	F18
(pre) 3 3 ~	KEY_F(55)	F19
(pre) 3 4 ~	KEY_F(56)	F20
(pre) 3 7 ~	KEY_F(13)	S−F1
(pre) 3 8 ~	KEY_F(14)	S−F2
(pre) 3 9 ~	KEY_F(15)	S−F3
(pre) 4 0 ~	KEY_F(16)	S−F4
(pre) 4 1 ~	KEY_F(17)	S−F5
(pre) 4 2 ~	KEY_F(18)	S−F6
(pre) 4 3 ~	KEY_F(19)	S−F7
(pre) 4 4 ~	KEY_F(20)	S−F8
(pre) 4 5 ~	KEY_F(21)	S−F9
(pre) 4 6 ~	KEY_F(22)	S−F10
(pre) 4 7 ~	KEY_F(23)	S−F11
(pre) 4 8 ~	KEY_F(24)	S−F12
(pre) 4 9 ~	KEY_F(25)	C−F1
(pre) 5 0 ~	KEY_F(26)	C−F2
(pre) 5 1 ~	KEY_F(27)	C−F3
(pre) 5 2 ~	KEY_F(28)	C−F4
(pre) 5 3 ~	KEY_F(29)	C−F5
(pre) 5 4 ~	KEY_F(30)	C−F6
(pre) 5 5 ~	KEY_F(31)	C−F7
(pre) 5 6 ~	KEY_F(32)	C−F8
(pre) 5 7 ~	KEY_F(33)	C−F9
(pre) 5 8 ~	KEY_F(34)	C−F10
(pre) 5 9 ~	KEY_F(35)	C−F11
(pre) 6 0 ~	KEY_F(36)	C−F12
(pre) A	KEY_UP	CURU
(pre) B	KEY_DOWN	CURD
(pre) C	KEY_RIGHT	CURR
(pre) D	KEY_LEFT	CURL
(pre) F	KEY_END	END
(pre) G	KEY_NextScreen	PGDN
(pre) H	KEY_HOME	HOME
(pre) I	KEY_PrevScreen	PGUP
(pre) L	KEY_InsertHere	INS
(pre) M	KEY_NUMENTER	NUMENTER
(pre) N	KEY_F(2)	F2
(pre) O	KEY_F(3)	F3
(pre) P	KEY_PF1	PF1
(pre) Q	KEY_PF2	PF2
(pre) R	KEY_PF3	PF3
(pre) S	KEY_PF4	PF4
(pre) T	KEY_F(8)	F8
(pre) U	KEY_F(9)	F9

(pre) V	KEY_F(10)	F10
(pre) W	KEY_F(11)	F11
(pre) X	KEY_F(12)	F12
(pre) Z	KEY_BACKTAB	S−TAB
(pre) l	KEY_PadComma	COMMA
(pre) m	KEY_PadMinus	MINUS
(pre) n	KEY_PadPeriod	NUMSTOP
(pre) o	KEY_Pad0	NUM0
(pre) p	KEY_Pad1	NUM1
(pre) q	KEY_Pad2	NUM2
(pre) r	KEY_Pad3	NUM3
(pre) s	KEY_Pad4	NUM4
(pre) t	KEY_Pad5	CENTER
(pre) u	KEY_Pad6	NUM6
(pre) v	KEY_Pad7	NUM7
(pre) w	KEY_Pad8	NUM8
(pre) x	KEY_Pad9	NUM9
(pre) y	KEY_NUMENTER	NUMENTER
(pre) 1 z	KEY_BTAB	S−TAB
(pre) 2 z	KEY_InsertHere	INS
(pre) 3 z	KEY_HOME	HOME
(pre) 4 z	KEY_END	END
(pre) 5 z	KEY_PrevScreen	PGUP
(pre) 6 z	KEY_NextScreen	PGDN
(pre) 1 9 5 z	KEY_UNDO	UNDO
(pre) 2 1 4 z	KEY_HOME	HOME
(pre) 2 1 5 z	KEY_UP	CURU
(pre) 2 1 6 z	KEY_PrevScreen	PGUP
(pre) 2 1 7 z	KEY_LEFT	CURL
(pre) 2 1 9 z	KEY_RIGHT	CURR
(pre) 2 2 0 z	KEY_END	END
(pre) 2 2 1 z	KEY_DOWN	CURD
(pre) 2 2 2 z	KEY_NextScreen	PGDN
(pre) 2 2 4 z	KEY_F(1)	F1
(pre) 2 2 5 z	KEY_F(2)	F2
(pre) 2 2 6 z	KEY_F(3)	F3
(pre) 2 2 7 z	KEY_F(4)	F4
(pre) 2 2 8 z	KEY_F(5)	F5
(pre) 2 2 9 z	KEY_F(6)	F6
(pre) 2 3 0 z	KEY_F(7)	F7
(pre) 2 3 1 z	KEY_F(8)	F8
(pre) 2 3 2 z	KEY_F(9)	F9
(pre) 2 3 3 z	KEY_F(10)	F10
(pre) 2 3 4 z	KEY_F(11)	F11
(pre) 2 3 5 z	KEY_F(12)	F12
(pre) 3 2 4 z	KEY_F(13)	S−F1
(pre) 3 2 5 z	KEY_F(14)	S−F2
(pre) 3 2 6 z	KEY_F(15)	S−F3
(pre) 3 2 7 z	KEY_F(16)	S−F4
(pre) 3 2 8 z	KEY_F(17)	S−F5
(pre) 3 2 9 z	KEY_F(18)	S−F6
(pre) 3 3 0 z	KEY_F(19)	S−F7
(pre) 3 3 1 z	KEY_F(20)	S−F8
(pre) 3 3 2 z	KEY_F(21)	S−F9

(pre) 3 3 3 z	KEY_F(22)	S−F10
(pre) 3 3 4 z	KEY_F(23)	S−F11
(pre) 3 3 5 z	KEY_F(24)	S−F12
(pre) 4 1 4 z	KEY_C_HOME	C−HOME
(pre) 4 1 5 z	KEY_C_CURU	C−CURU
(pre) 4 1 6 z	KEY_C_PGUP	C−PGUP
(pre) 4 1 7 z	KEY_C_CURL	C−CURL
(pre) 4 1 9 z	KEY_C_CURR	C−CURR
(pre) 4 2 0 z	KEY_C_END	C−END
(pre) 4 2 1 z	KEY_C_CURD	C−CURD
(pre) 4 2 2 z	KEY_C_PGDN	C−PGDN
(pre) 4 2 3 z	KEY_PadComma	COMMA
(pre) 4 2 4 z	KEY_F(25)	C−F1
(pre) 4 2 5 z	KEY_F(26)	C−F2
(pre) 4 2 6 z	KEY_F(27)	C−F3
(pre) 4 2 7 z	KEY_F(28)	C−F4
(pre) 4 2 8 z	KEY_F(29)	C−F5
(pre) 4 2 9 z	KEY_F(30)	C−F6
(pre) 4 3 0 z	KEY_F(31)	C−F7
(pre) 4 3 1 z	KEY_F(32)	C−F8
(pre) 4 3 2 z	KEY_F(33)	C−F9
(pre) 4 3 3 z	KEY_F(34)	C−F10
(pre) 4 3 4 z	KEY_F(35)	C−F11
(pre) 4 3 5 z	KEY_F(36)	C−F12
(pre) 1 9 2 z	KEY_F(49)	F13
(pre) 1 9 3 z	KEY_F(50)	F14
(pre) 2 9 2 z	KEY_F(57)	S−F13
(pre) 2 9 3 z	KEY_F(58)	S−F14
(pre) [A	KEY_F(1)	F1
(pre) [B	KEY_F(2)	F2
(pre) [C	KEY_F(3)	F3
(pre) [D	KEY_F(4)	F4
(pre) [E	KEY_F(5)	F5
(pre) 0 q	KEY_PadComma	COMMA
(pre) 1 q	KEY_F(1)	F1
(pre) 2 q	KEY_F(2)	F2
(pre) 3 q	KEY_F(3)	F3
(pre) 4 q	KEY_F(4)	F4
(pre) 5 q	KEY_F(5)	F5
(pre) 6 q	KEY_F(6)	F6
(pre) 7 q	KEY_F(7)	F7
(pre) 8 q	KEY_F(8)	F8
(pre) 9 q	KEY_F(9)	F9
(pre) 1 0 q	KEY_F(10)	F10
(pre) 1 1 q	KEY_F(11)	F11
(pre) 1 2 q	KEY_F(12)	F12
(pre) 1 3 q	KEY_F(13)	S−F1
(pre) 1 4 q	KEY_F(14)	S−F2
(pre) 1 5 q	KEY_F(15)	S−F3
(pre) 1 6 q	KEY_F(16)	S−F4
(pre) 1 7 q	KEY_F(17)	S−F5
(pre) 1 8 q	KEY_F(18)	S−F6
(pre) 1 9 q	KEY_F(19)	S−F7
(pre) 2 0 q	KEY_F(20)	S−F8

(pre) 2 1 q	KEY_F(21)	S−F9
(pre) 2 2 q	KEY_F(22)	S−F10
(pre) 2 3 q	KEY_F(23)	S−F11
(pre) 2 4 q	KEY_F(24)	S−F12
(pre) 2 5 q	KEY_F(25)	C−F1
(pre) 2 6 q	KEY_F(26)	C−F2
(pre) 2 7 q	KEY_F(27)	C−F3
(pre) 2 8 q	KEY_F(28)	C−F4
(pre) 2 9 q	KEY_F(29)	C−F5
(pre) 3 0 q	KEY_F(30)	C−F6
(pre) 3 1 q	KEY_F(31)	C−F7
(pre) 3 2 q	KEY_F(32)	C−F8
(pre) 3 3 q	KEY_F(33)	C−F9
(pre) 3 4 q	KEY_F(34)	C−F10
(pre) 3 5 q	KEY_F(35)	C−F11
(pre) 3 6 q	KEY_F(36)	C−F12
(pre) 1 3 9 q	KEY_InsertHere	INS
(pre) 1 5 0 q	KEY_PrevScreen	PGUP
(pre) 1 4 6 q	KEY_END	END
(pre) 1 5 4 q	KEY_NextScreen	PGUP
ESC P	KEY_PF1	PF1
ESC Q	KEY_PF2	PF2
ESC R	KEY_PF3	PF3
ESC S	KEY_PF4	PF4

Where (pre) is either:

```
        ESC [ − (0x1B 0x5B) or
        ESC O − (0x1B 0x4F) or
        ESC ? − (0x1B 0x3F) or
        CSI − (0x9B)
```

Generated on: 2 Nov 2002

Return to Table of Contents

APPENDIX 3 − POINTER DEVICE SUPPORT IN THE

This appendix describes the default behaviour of supported pointer devices when running THE; usually a mouse.

SYNTAX:

Mouse "keys" have the following syntax:

[a−]bcB IN window	Description
[a−]	is a keyboard modifier, the same as keyboard keys. The 'a' can be one of S,C or A for SHIFT, CONTROL and ALT respectively. The '−' is mandatory.
b	is the mouse action. This can be one of P, R, C, 2 or D for PRESS, RELEASE, CLICK, DOUBLE_CLICK and DRAG respectively.
c	is the mouse button. This can be one of L, R or M for LEFT, RIGHT and MIDDLE respectively.
B	stands for BUTTON
IN	mandatory keyword
window	the area of the screen in which the mouse event is to take place. The valid values for this are: Filearea, Cmdline, Idline, Arrow, PRefix, STatarea and Divider

EXAMPLES:

Mouse Event	Description
S−PLB IN PR	Left mouse button is pressed, while holding down the SHIFT key, and the mouse positioned in the prefix area.
2RB IN I	Right mouse button double−clicked in idline.

DEFAULTS:

Mouse Event (PDCurses)	Command(s)
PLB in FILEAREA	cursor MOUSE
S−PLB in FILEAREA	cursor MOUSE#reset BLOCK#mark LINE
C−PLB in FILEAREA	cursor MOUSE#reset BLOCK#mark BOX
S−PRB in FILEAREA	cursor MOUSE#mark LINE
C−PRB in FILEAREA	cursor MOUSE#mark BOX

PRB in FILEAREA	cursor MOUSE#sos MAKECURR
S−DLB in FILEAREA	cursor MOUSE#mark LINE
C−DLB in FILEAREA	cursor MOUSE#mark BOX
2LB in FILEAREA	cursor MOUSE#sos EDIT
PLB in PREFIX	cursor MOUSE
PRB in PREFIX	cursor MOUSE#sos MAKECURR
2LB in PREFIX	cursor MOUSE#sos EDIT
PLB in CMDLINE	cursor MOUSE
PLB in STATAREA	status
PLB in IDLINE	xedit
PRB in IDLINE	xedit −
PLB in DIVIDER	screen 1

Mouse Event (ncurses)	**Command(s)**
CLB in FILEAREA	cursor MOUSE
CRB in FILEAREA	cursor MOUSE#sos MAKECURR
2LB in FILEAREA	cursor MOUSE#sos EDIT
CLB in PREFIX	cursor MOUSE
CRB in PREFIX	cursor MOUSE#sos MAKECURR
2LB in PREFIX	cursor MOUSE#sos EDIT
CLB in CMDLINE	cursor MOUSE
CLB in STATAREA	status
CLB in IDLINE	xedit
CRB in IDLINE	xedit −
CLB in DIVIDER	screen 1

NOTES:

1. In the X11 port, pressing the middle mouse button anywhere within the THE window, will result in the contents of the X selection being pasted where the text cursor is currently displayed. If you hold down any modifier key, such as SHIFT or CONTROL, that sequence is passed to THE and can be mapped. Therefore; PMB or RMD is NOT able to be used in THE, but C−PMB, C−RMB, S−PMB etc. can.

2. The button action DOUBLE_CLICKED will always be preceded by a CLICK action. This is the case for THE with ncurses or PDCurses. Therefore itis usually a good idea to have the CLICK action assigned to a cursor positioning command if the DOUBLE_CLICK for the same button is used.

Return to Table of Contents

APPENDIX 4 − SYNTAX HIGHLIGHTING IN THE

This appendix contains details on syntax highlighting in THE. Syntax highlighting is the mechanism by which different tokens within a file; usually containing source code, are displayed in different colours.

The model THE uses for its syntax highlighting is based on the model used by KEDIT for Windows from Mansfield Software. This model is extremely configurable and flexible. While most of the KEDIT features are implemented, THE also adds a couple of other features that make the syntax highlighting even better.

This appendix concentrates on the format of THE language definition files. For a description of the commands that manipulate other aspects of syntax highlighting in THE, see the descriptions of the following commands: SET AUTOCOLOR , SET COLORING , SET ECOLOR , SET PARSER .

Performance Impact

Syntax highlighting in an editor comes at a cost; reduced performance.

Because of the extra processing required to determine which characters are displayed in which colours, displaying the screen is slower. As THE recalculates the display colours after every displayable key is pressed, then you may notice a reduction in responsiveness.

The more features that are specified in a TLD, the slower the syntax highlighting will be. To dynamically turn on or off the application of some headers within a TLD file, see the SET HEADER command.

For those languages that allow paired comments (ie they can span multiple lines) performance is impacted even more. This is because THE has to determine if the lines being displayed are within one of these multi−line comment pairs which may start before the first displayed line.

THE will incorrectly display syntax highlighting in certain circumstances. This is because THE does not fully parse the complete file to determine the correct colours; that would be too slow. Instead, THE checks the currently displayed lines and determines the syntax highlighting based on these lines.

Where THE will get syntax highlighting wrong:

If all displayed lines are within a multi−line comment block and neither the starting comment token nor the ending comment token are displayed. THE will treat the displayed lines as code.

When the starting or ending comment tokens for multi−line comments are part of a language string.

Also bear in mind that excluding large portions of the file with ALL, will dramatically slow down checking of multi−line comments.

File Extensions Vs Magic Numbers

A THE extension to the KEDIT syntax highlighting model is support for magic numbers . (See SET AUTOCOLOR for more details). For the default parser s, where there might be a conflict between setting syntax highlighting based on a file extension or a magic number , the file extension mapping takes precedence.

THE Language Definition Files

THE Language Definition Files usually have a file extension of .tld. THE comes with a small number of sample TLD files. Look at these files in conjunction with the following descriptions to fully understand how to write your own TLD files.

TLD files consist of several sections identified by header lines. Header lines start with a colon in column one. Items within the particular header are listed on separate lines after the header to which they apply. Blank lines are ignored, and so are comments (* as first non−blank). Each item that can be repeated occurs on a separate line. The above definition of what a TLD file looks like is expressed in the TLD file; tld.tld.

Many items in a TLD are specified as a regular expression (RE). THE supports a number of RE syntaxes for targets. All REs specified in a TLD are parsed using the EMACS syntax. For details of RE usage in THE see Appendix 7.

The purpose of each header and the valid contents are explained below.

:identifier

This section specifies, using a regular expression how a keyword in the language is defined. The only item line contains three regular expressions separated by space characters.
Syntax:
first_char_re other_char_re [last_char_re]
Meaning of options:
first_char_re
This regular expression specifies the valid characters that an identifier can begin with.
other_char_re
This regular expression specifies the valid characters that the remainder of characters in an identifier can consist of.
last_char_re
This regular expression is optional. If specified, it states the valid characters that an identifier can end with.

:case

This section defines whether the case of letters that make up identifiers in the language are case−sensitive or not. Only one of the items below can be included.
Syntax:
RESPECT | IGNORE
Meaning of options:
respect
case is relevant. The keywords *if* , *IF* and *If* are different.
ignore
case is irrelevant. The keywords *if* and *IF* are treated as the same identifier.

:option

This section specifies different options that can affect other sections. The options below can all be included in the one TLD.
Syntax:
REXX
PREPROCESSOR char
FUNCTION char BLANK | NOBLANK [DEFAULT ALTernate x]
Meaning of options:
rexx

specifies special processing for Rexx. eg. Functions defined in the :functions section, are also highlighted if preceeded by CALL.
preprocessor char
languages like C that have preprocessor identifiers usually begin with a special character (specified by *char*) to differentiate these types of keywords from others.
function char blank | noblank [default alternate x]
this option is used to identify how keywords specified in the :function section are identified. *char* specifies the character that is used to start a function, usually *(* . The *blank* or *noblank* argument determines if blank characters can appear between the function identifier and the function start character. eg a Rexx function call must be written without blanks between the function name and the function start character: *word(* . In C *word (* or *word(* are both valid syntax for a function call. The optional "default alternate x" specifies the color in which functions that are NOT specified in the :function section are to be displayed. See the explanation of alternate colors in the :function section. Without "default alternate x", the color of unknown functions is not changed.

:number

This section specifies the format of numbers in the language. Most languages use a small number of generic types of numbers.
Syntax:
REXX | C | COBOL
Meaning of options:
ECOLOR Value: Numbers are displayed in the colour specified with ECOLOUR *C* .

:string

This section specifies how strings within the language are defined. Multiple values may be specified, as many languages use both single and double quotes.
Syntax:
SINGLE [BACKSLASH] | DOUBLE [BACKSLASH]
Meaning of options:
single
Specifies that the language uses single quotes to identify a string.
double
Specifies that the language uses double quotes to identify a string.
backslash
Some languages require a backslash character immediately preceding either a single or double quote to allow the quote to be included in the string.
ECOLOR Character:
For complete strings, the ECOLOUR character used is *B* . For incomplete strings, the ECOLOUR character used is *S* .

:comment

This section specifies the format of comments. Both paired and line comments can be specified, as can multiple occurrences of each.
Syntax:
PAIRED open_string close_string [NEST | NONEST]
LINE comment_string ANY | FIRSTNONBLANK | COLUMN n
Meaning of options:
paired
These types of comments can span multiple lines. They have an opening string and a closing string.
open_string
This defines the string that opens a paired comment.

close_string
This defines the string that closes a paired comment.
nest
Some languages allow paired comments to be nested. (not implemented)
nonest
Defining this indicates that the language does not allow nesting of paired comments. The effect of this option will result in the first close_string to end the paired comment no matter how many open_string occurrences there are. (not implemented)
line
These type of comments cannot span multiple lines. Everything on the line after the comment_string is considered part of the comment.
comment_string
The string that defines a line comment.
any
For line comments, this indicates that the comment_string can occur anywhere on the line, and all characters following it are part of the comment.
firstnonblank
For line comments, this indicates that the comment_string can only occur as the first non−blank of the line.
column n
For line comments, this indicates that the comment_string must start in the specified column.
ECOLOR Character:
Comments are displayed in the colour specified with ECOLOUR *A* .

:keyword

This section specifies all of the identifiers that are to be considered language keywords. You must specific the :identifier section in the TLD file before the :keyword section.
Syntax:
keyword [ALTernate x] [TYPE x]
Meaning of options:
keyword
This specifies the string that is considered to be a language keyword.
alternate x
All keywords are displayed in the same colour, unless you use this option to specify a different colour. In KEDIT there are 9 alternate colours that can be used; ECOLOUR 1 through 9. In THE any ECOLOUR character can be used as an alternate colour. *alternate* can be abbreviated to *alt* .
type x
(not implemented)
ECOLOR Character:
Unless overridden by the *alternate* option, the keyword is displayed in the colour specified with ECOLOUR *D* .

:function

This section specifies all of the identifiers that are to be considered functions. Normally this is used for those functions that are builtin into the language, but can be any identifier. You specify the function identifier without the function char specified in the :option section. You must specify the :option and the :identifier sections in the TLD file before the :function section.
Syntax:
function [ALTernate x]
Meaning of options:
function
This specifies the string that is considered to be a language function.
alternate x

All functions are displayed in the same colour, unless you use this option to specify a different colour. In KEDIT there are 9 alternate colours that can be used; ECOLOUR 1 through 9. In THE any ECOLOUR character can be used as an alternate colour. *alternate* can be abbreviated to *alt* .
ECOLOR Character:
Unless overridden by the *alternate* option, the function is displayed in the colour specified with ECOLOUR *V* .

:header

This section specifies the format of headers. Headers are lines within a file that begin with a particular string and usually identify different parts of the file. They are similar to labels.
Syntax:
LINE header_string ANY | FIRSTNONBLANK | COLUMN n
Meaning of options:
header_string
The string that defines a header.
any
This indicates that the header_string can occur anywhere on the line, and all characters following it are part of the header.
firstnonblank
This indicates that the header_string can only occur as the first non−blank of the line.
column n
This indicates that the header_string must start in the specified column.
ECOLOR Character:
Headers are displayed in the colour specified with ECOLOUR *G* .

:label

This section specifies the format of labels. Labels are lines within a file that end with a particular string. They are similar to headers.
Syntax:
DELIMITER label_string ANY | FIRSTNONBLANK | COLUMN n
COLUMN n
Meaning of options:
label_string
The string that defines a label.
any
This indicates that the label_string can occur anywhere on the line, and all characters up to it are part of the label.
firstnonblank
This indicates that the label_string can only occur as the first non−blank of the line.
column n
As part of a DELIMITER label, this indicates that the label_string must start in the specified column. If specified by itself, then the label does not require any special delimiter; the non−keyword that starts in the specified column is regarded as a label.
ECOLOR Character:
Labels are displayed in the colour specified with ECOLOUR *E* .

:markup

This section specifies the delimiters for a markup tag, and optionally the delimiters for references within a markup language.
Syntax:
TAG tag_start tag_end [REFERENCE ref_start ref_end]

Meaning of options:
tag_start
The character that specifies the start of a markup tag.
tag_end
The character that specifies the end of a markup tag.
ref_start
The character that specifies the start of a markup reference.
ref_end
The character that specifies the end of a markup reference.
ECOLOR Character:
Tags are displayed in the colour specified with ECOLOUR *T* . References are displayed in the colour spceified with ECOLOUR *U* .

:match

(Not implemented yet)

:column

This section specifies the range of columns in your file which is to have syntax highlighting applied. For example, columns 1−6 and beyond column 72 in a COBOL source file should be excluded from being parsed. Any number of EXCLUDE clauses are allowed. Note. Not all syntax checking respects excluded columns at this stage.
Syntax:
EXCLUDE first_column last_column [ALTernate x]
Meaning of options:
first_column
The first column to be excluded
last_column
The last column to be excluded. * can be used to specify to the end of the line.
alternate x
All excluded characters are displayed in the same colour, unless you use this option to specify a different colour. In KEDIT there are 9 alternate colours that can be used; ECOLOUR 1 through 9. In THE any ECOLOUR character can be used as an alternate colour. *alternate* can be abbreviated to *alt* .
ECOLOR Character:
Unless overridden by the *alternate* option, the excluded characters are displayed in the colour specified with COLOUR FILEAREA.

:postcompare

This section specifies items that are checked for after all other syntax checking has been completed. This can be useful if you want to allow user−defined datatypes or other code to be displayed in different colours.
Syntax:
CLASS re [ALTernate x]
TEXT string [ALTernate x]
Meaning of options:
re
This regular expression specifies the text to be highlighted.
string
This indicates the literal string to be highlighted.
alternate x
All matched postcompare characters are displayed in the same colour, unless you use this option to specify a different colour. In KEDIT there are 9 alternate colours that can be used; ECOLOUR 1 through 9. In THE any ECOLOUR character can be used as an alternate colour. *alternate* can be abbreviated to *alt* .

ECOLOR Character:
Unless overridden by the *alternate* option, the matched characters are displayed in the colour specified with ECOLOUR *D* .

Builtin Parsers

THE includes a number of builtin syntax highlighting parser s. The following table lists the default parser s and the files they apply to:

Parser	Filemasks	"Magic Number"
REXX	*.rex *.rexx *.cmd *.the .therc	rexx regina rxx
C	*.c *.h *.cc *.hpp *.cpp	
SH		sh ksh bash zsh
TLD	*.tld	
HTML	*.html *.htm	

A Rexx macro is provided; tld2c.rex, to convert a .tld file into the C code that can be embedded in default.c. This enables you to configure THE with the default parser s that are more applicable for you.

Return to Table of Contents

APPENDIX 5 − DEFAULT STATUS SETTINGS IN THE

In case you never want to write macros or profiles you can safely return to Table of Contents :−)

This appendix describes the default status settings in THE in its various compatibility modes. You can inspect or save the current status of your THE−session with the `status` command. A single status information can be obtained with the `query` command. Within a macro you can use either the `extract` command or the implied extract method. You can set or change a status variable either with the `set` or `modify` command (see detailed or quick reference).

***NOTE**: Depending on* `set compat` *the following commands may return **different values** within macros (see also history for 2.3 and 2.5):*
`CURLINE`, `LINE`, `LINEFLAG`, `POINT`, `RING`, `SELECT`!

Topics:

- General defaults for status variables
- General defaults for tabs settings
- Settings depending on operating system
- Settings depending on current file
- Settings depending on current line/column
- Settings depending initially on external conditions
- Other status variables

Meaning of the flags in the following tables:

1. `F`: file scope, `G`: global scope (applies to all files in the ring); Note: when you open a new file for editing this file may inherit file specific settings from the current file (exception: `msgmode`, see also set reprofile).
2. `Q`: can be used with `query` command
3. `E`: can be used with `extract` command
4. `M`: can be changed with `set` or `modify` command
5. `S`: can be obtained by `status` command
6. `R`: can be obtained by implied extract within REXX macros
7. `P`: used by `preserve`/`restore` command

`?`: not yet determined
`−`: not available
lower case: not generally applicable, please refer to documentation

General defaults for status variables:

variable	THE	KEDIT	XEDIT	flags
arbchar	OFF $?	OFF $?	OFF $?	`FQEMSRP`
autosave	OFF	OFF	OFF	`FQEMS−P`
backup	KEEP	OFF	OFF	`FQEMS−P`

beep	OFF	OFF	OFF	GQEMSR-
case	MIXED IGNORE RESPECT RESPECT	MIXED IGNORE RESPECT RESPECT	UPPER IGNORE RESPECT RESPECT	FQEMS-P
clearerrorkey	*	*	*	GQEMSR-
clearscreen	OFF	ON	ON	GQEMSR-
clock	ON	ON	OFF	GQEMSR-
cmdarrows	RETRIEVE	TAB	TAB	GQEMS--
cmdline	BOTTOM	BOTTOM	BOTTOM	FQEMSrP
compat	THE THE THE	KEDIT KEDIT KEDIT	XEDIT XEDIT XEDIT	GQEMSR-
curline	M	M	M	FqEmsRP
cursorstay	ON	ON	ON	GQEMSR-
defsort	NAME ASCENDING	NAME ASCENDING	NAME ASCENDING	GQEMSR-
dirinclude	*	*	*	GQEMSR-
display	0 0	0 0	0 0	FQEMSRP
fullfname	ON	ON	ON	FQEMSR-
hex	OFF	OFF	OFF	F-EMSRP
hexdisplay	ON	OFF	OFF	GQEMSR-
hexshow	OFF 7	OFF 7	OFF 7	FQEMSRP
highlight	OFF	OFF	OFF	FQEMSRP
idline	ON	ON	ON	FQEMSRP
impcmscp	ON	ON	ON	G--m--?
impmacro	ON	OFF	ON	FQEMSRP
impos	ON	OFF	ON	GQEMS-P
inputmode	LINE	LINE	LINE	FQEMS-P
insertmode	OFF	OFF	OFF	FQEMSR-
linend	OFF #	OFF #	ON #	GQEMS-P
macro	OFF	OFF	OFF	GQEMSRP
macroext	the	the	the	GQEMSR-
margins	1 72 +0	1 72 +0	1 72 +0	FQEMS-P
msgline	ON 2 5 OVERLAY	ON 2 5 OVERLAY	ON 2 2 OVERLAY	FQEMSRP
msgmode	ON	ON	ON	FQEMSRP
newlines	ALIGNED	ALIGNED	ALIGNED	FQEMSRP
nondisp	#	#	"	GQEMSR-
number	ON	OFF	OFF	FQEMSRP
position	ON	ON	ON	FqEmsRP
prefix	ON LEFT 6 0	OFF	ON LEFT 6 1	FqEmsrP
reprofile	OFF	OFF	OFF	GQEMSR-
rexxoutput	DISPLAY 1000	DISPLAY 1000	DISPLAY 1000	GQEMSR-
scale	OFF M+1	OFF M+1	ON M+1	FQEMSRP
scope	DISPLAY	DISPLAY	DISPLAY	FQEMSRP

screen	1 HORIZONTAL	1 HORIZONTAL	1 HORIZONTAL	GQEMSR-
shadow	ON	ON	ON	FQEMSRP
span	OFF Blank 2	OFF Blank 2	OFF Blank 2	?--m--?
spill	OFF	OFF	OFF	?--m--?
statopt	ON NBFILE.1 13 0 Files= ON WIDTH.1 23 0 Width=	ON NBFILE.1 13 0 Files= ON WIDTH.1 23 0 Width=	ON NBFILE.1 13 0 Files= ON WIDTH.1 23 0 Width=	G--m---
statusline	BOTTOM	BOTTOM	BOTTOM	FQEMSR-
stay	ON	ON	OFF	FQEMSRP
tabkey	TAB CHARACTER	TAB CHARACTER	TAB CHARACTER	GQEMSR-
tabline	OFF −3	OFF −2	OFF −3	FQEMSRP
tabsin	OFF 8	OFF 8	OFF 8	GQEMSR-
tabsout	OFF 8	OFF 8	OFF 8	GQEMSRP
timecheck	ON	ON	ON	FQEMSR-
trunc	*	*	*	?--m--?
typeahead	OFF	OFF	OFF	GQEMSR-
undoing	ON	ON	ON	FQEMSR-
untaa	OFF	OFF	OFF	GQEMSR-
word	NONBLANK	NONBLANK	NONBLANK	FQEMSRP
wordwrap	OFF	OFF	OFF	FQEMS-P
wrap	OFF	OFF	OFF	FQEMSR-

Back to topics list.

General defaults for tabs settings:

compatibility	**tabs**
THE:	1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137
	145 153 161 169 177 185 193 201 209 217 225 233 241 249
KEDIT:	1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137
	145 153 161 169 177 185 193 201 209 217 225 233 241 249
XEDIT:	1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55
	58 61 64 67 70 73 76 79 82 85 88 91 94

Back to topics list.

Settings depending on operating system:

variable	**flags**
eolout	FQEMSRP
etmode	GQEMSR-

monitor	GQE-SR-
mouse	GQEMSR-
printer	GQEMSR-
slk	F--m---
terminal	GQE-S--
xterminal	GQEMSR-

Back to topics list.

Settings depending on current file:

variable	flags
alt	FQEmSR-
fext	FQEM-R-
filename	FQEM-R-
filestatus	FQE-SR-
fmode	FQEM-R-
fname	FQEM-R-
fpath	FQEM-R-
ftype	FQEM-R-
point	FqE----
reserved	FqE-sR-
size	FQE-SR-

Back to topics list.

Settings depending on current line/column:

variable	flags
block	GQE----
column	FQE--R-
curline	FqEmsRp
cursor	FQE--R-
dirfileid	f-E--R-
eof	FQE-SR-
field	Fqe--R-
length	FQE-SR-
line	FQE-SR-
lineflag	FQEMSR-
select	FQE-SR-
tof	FQE-SR-

vershift	FQE−SR−

Back to topics list.

Settings depending initially on external conditions:

variable	flags
lscreen	FQE−−R−
macropath	GQEM−R−
verify	FQEMSR−
width	GQE−SR−
zone	FQEMSRP

Back to topics list.

Other status variables:

variable	flags
colour	FqeM−r−
getenv	G−e−−R−
lastkey	G−E−−R−
lastmsg	G−E−−R−
lastrc	GQE−−R−
nbfile	GQE−SR−
pending	F−E−−−−
ring	GQE−−R−
showkey	G−E−−−−
version	GQE−SR−

Back to topics list.

Last changed on: 09 Nov 1998 by Franz−Josef Wirtz<fw@ecotopia.oche.de>

Return to Table of Contents

APPENDIX 6 − THE BY TASKS

Back to to Table of Contents

This appendix gives an overview, how general concepts are reflected by various THE commands, settings, key bindings etc.

Topics:

- Getting Help
- Querying/ Changing Status
- Moving Around
- Scrolling
- Inserting, Deleting and Recovering
- Searching, Replacing, Bookmarking
- Block Marked Regions
- File Handling
- Printing
- Folding (Partial Views)
- Windows
- Operating System
- Command Line
- Prefix Area
- Macros
- Keys
- Miscellaneous

Notation for keys: C− control modifier, A− Alt modifier, S− shift modifier, X− either A− or C− modifier (depending on operating system)

Note on set commands and functions: functions, which can refer to status infos which can be modified directly with set are not mentioned explicitly.

Getting Help

keys	command	set command	sos command	function	prefix
F1 (help)	help	set/extract beep set/extract clock set/extract hexdisplay set/extract hexshow set/extract highlight set/extract idline set/extract msgline set/extract msgmode set/extract number set/extract position set/extract prefix set/extract reserved			TABL SCALE

		set/extract rexxoutput			
		set/extract scale			
		set slk			
		set statopt			
		set/extract statusline			
		set/extract tabline			

Back to topics list.

Querying/ Changing Status

keys	command	set command	sos command	function	prefix
	editv	set/extract clearerrorkey		lastmsg.1()	
	extract	extract getenv		lastrc.1()	
	modify	extract lastkey			
	preserve	extract lastmsg			
	query	extract lastrc			
	restore	set/extract msgline			
	set	set/extract msgmode			
	showkey	extract showkey			
	status	set statopt			
		set/extract statusline			

Back to topics list.

Moving Around

keys	command	set command	sos command	function	prefix
PGUP (backward 1)	backward	extract column	sos blockend	after()	
PGDN (forward 1)	bottom	extract curline	sos blockstart	before()	
C−PGUP (backward *)	cfirst	extract cursor	sos bottomedge	bottomedge()	
C−PGDN (forward *, top?)	clast	set/extract cursorstay	sos current	end()	
CURU (cursor up)	cursor	extract eof	sos cursoradj	eof()	
CURD (cursor down)	down	extract line	sos endchar	first()	
CURL (cursor left)	forward	set/extract margins	sos firstchar	focuseof()	
CURR (cursor right)	next	extract position	sos firstcol	focustof()	
C−CURL,F18 (sos tabwordb)	top	set/extract tabkey	sos lastcol	incommand()	
C−CURR,F19 (sos tabwordf)	up	set/extract tabs	sos leftedge	inprefix()	
HOME (cursor home)		extract tof	sos marginl	leftedge()	
END (sos startendchar)		set/extract word	sos marginr	rightedge()	
TAB,C−I (sos tabf)			sos parindent	tof()	
S−TAB (sos tabb)			sos prefix	topedge()	
C−A (sos cursoradj)			sos qcmnd	column.1()	
			sos rightedge	curline.2()	
			sos settab	cursor.3()	
			sos startendchar	cursor.4()	
			sos tabb	cursor.7()	
			sos tabf	cursor.8()	
			sos tabfieldb	line.1()	
			sos tabfieldf	position.2()	
			sos tabwordb	position.3()	

			sos tabwordf sos topedge		

Back to topics list.

Scrolling

keys	command	set command	sos command	function	prefix

F5 (sos makecurr)	backward
	forward
	left
	rgtleft
	right
		set/extract cursorstay	sos makecurr	verone()
				vershift.1()
					/ (sos makecurr)

Back to topics list.

Inserting, Deleting and Recovering

keys	command	set command	sos command	function	prefix

ESC,C−Q (sos undo)
F4 (sos addline 1)
F8 recover
F9 (sos delline 1)
F10 (splitjoin)
F11 (controlchar)
BKSP... (sos delback)
INS (insertmode toggle)
DEL, C−H... (sos delchar)
C−END, C−E (sos delend)
X−F (join aligned)
X−S (split aligned)
X−W (sos delword)
	add
	cappend
	cdelete
	cinsert
	compress
	controlchar
	coverlay
	delete
	duplicate
	enter
	expand
	input
	join
	recover
	shift
	split
	spltjoin
	text
		set/extract case
		extract curline
		set/extract inputmode
		set/extract insertmode
		extract lineflag
		set/extract margins
		set/extract newlines
		set/extract tabs
		set/extract tabsin
		set/extract tabsout
		set/extract undoing
		set/extract word
		set/extract wordwrap
			sos addline
			sos delback
			sos delchar
			sos delend
			sos delline
			sos delword
			sos instab
			sos lineadd
			sos linedel
			sos undo
				modifiable()
				curline.3()
				lineflag.1()
				lineflag.2()
					" (duplicate)
					> (shift right)
					< (shift left)
					a (add)
					c (copy)
					d (delete)
					f (insert following)
					i (add)
					m (move)
					p (insert prior)

Back to topics list.

Searching, Replacing, Bookmarking

keys	command	set command	sos command	function	prefix

S−F1 (locate)
F7 (cmatch)
C−F11 (point .a)
C−F12 (locate .a)
	all
	change
	clocate
	cmatch
	creplace
	find
		set/extract arbchar
		set/extract case
		set/extract hex
		extract lineflag
		set/extract point
		set/extract scope
				lineflag.2()
				valid_target()
					.a (set point)

	findup locate lowercase nfind nfindup replace schange sort toascii uppercase	set span set spill set/extract stay set trunc set/extract untaa set/extract wrap set/extract zone			

Back to topics list.

Block Marked Regions

keys	command	set command	sos command	function	prefix

X−K (copy block)
X−B (mark box)
X−C (copy block reset)
X−D (duplicate 1 block)
X−F (fillbox)
X−G (delete block)
X−L (mark line)
A−M, C−V (move block reset)
X−O (overlay box)
X−U (reset block)
	copy
	fillbox
	lowercase
	mark
	move
	overlaybox
	put
	putd
	reset
	uppercase
		set/extract insertmode
		(?)
		set/extract word
		extract block
			sos blockend
			sos blockstart
			sos pastecmdline
				block()
				inblock()
				block.1()
				block.2()
				block.3()
				block.4()
				block.5()
				block.6()
					"" (duplicate)
					>> (shift right)
					<< (shift left)
					cc (copy)
					dd (delete)
					f (insert following)
					mm (move)
					p (insert prior)
					xx (all BLOCK)

Back to topics list.

File handling

keys	command	set command	sos command	function	prefix

F3 (file)
S−F3,F13 (qquit)
X−X (sos edit)
	cancel
	ccancel
	directory
	edit
	ffile
	file
	get
	kedit
	ls
	put
	putd
	qquit
	quit
	save
	ssave
	the
	xedit
		set/extract alt
		set/extract autosave
		set/extract backup
		extract block
		set/extract defsort
		extract dirfileid
		set/extract dirinclude
		set/extract eolout
		set/extract fext
		set/extract filename
		extract filestatus
		set/extract fmode
		set/extract fname
		set/extract fpath
		extract ftype
		set/extract fullfname
		set/extract idline
		extract nbfile
			sos edit	alt()
				dir()
				block.6()
				dirfileid.1()
				dirfileid.2()
				nbfile.1()
				ring.1()
				ring.n()
				size.1()	

		extract ring			
		extract size			
		set/extract tabs			
		set/extract tabsin			
		set/extract tabsout			
		set/extract timecheck			
		set/extract undoing			
		set/extract width			

Back to topics list.

Printing

keys	command	set command	sos command	function	prefix
	print	set/extract printer			

Back to topics list.

Folding (Partial Views)

keys	command	set command	sos command	function	prefix
	all	set/extract display		lineflag.3()	s (all)
		set/extract scope		shadow()	x (all LINE)
		set/extract select			xx (all BLOCK)
		set/extract shadow			
		set/extract verify			
		set/extract zone			
		extract lineflag			

Back to topics list.

Windows

keys	command	set command	sos command	function	prefix
F2 (nextwindow)	cursor	set/extract clock	sos bottomedge	bottomedge()	/ (make current)
F12,PLUS... (tabpre)	edit	set/extract colour	sos execute	leftedge()	
HOME (cursor home)	kedit	set/extract compat	sos leftedge	inprefix()	TABL
C−R (redraw)	nextwindow	set/extract cmdline	sos pastecmdline	incommand()	SCALE
	prevwindow	set/extract curline	sos prefix	rightedge()	
	redraw	extract cursor	sos qcmnd	shadow()	
	refresh	set/extract hexdisplay	sos rightedge	topedge()	
	tabpre	set/extract hexshow	sos tabfieldf	curline.2()	
	the	set/extract idline	sos tabfieldb	cursor.1()	
	xedit	extract lscreen	sos topedge	cursor.2()	
		set/extract position		cursor.5()	
		set/extract prefix		cursor.6()	
		set pscreen		lscreen.1()	
		set/extract reserved		lscreen.2()	
		set/extract screen		lscreen.3()	

		set/extract scale		lscreen.4()	
		set/extract shadow		lscreen.5()	
		set slk		lscreen.6()	
		set/extract statopt		vershift.1()	
		set/extract statusline			
		set/extract tabkey			
		set/extract tabline			
		set/extract verify			
		extract vershift			

Back to topics list.

Operating System

keys	command	set command	sos command	function	prefix
X−X (sos edit)	!	set/extract defsort	sos edit	version.3()	
C−Z (suspend)	directory	set/extract dirinclude		run_os()	
	dos	extract getenv			
	dosnowait	set impcmscp			
	dosquiet	set/extract impos			
	ls	extract version			
	os				
	osnowait				
	osquiet				
	osredir				
	suspend				

Back to topics list.

Command Line

keys	command	set command	sos command	function	prefix
ESC,C−Q (sos undo)	?	set/extract cmdarrows	sos execute	command()	
F6 (?)	=	set/extract cmdline	sos pastecmdline	incommand()	
F12,PLUS... (tabpre)	cmsg	extract lastrc	sos qcmnd	lastrc.1()	
HOME (cursor home)	command	set/extract linend	sos tabfieldf		
ENTER (enter)	enter	set/extract macro	sos tabfieldb		
	input	set/extract scope	sos undo		
	nomsg	set synonym			
	readv cmdline	set/extract tabkey			
	repeat	set trunc			
	rexx	set/extract zone			
	sos				

Back to topics list.

Prefix Area

keys	command	set command	sos command	function	prefix

ENTER,C−M (enter)
F12,PLUS... (tabpre)
HOME (cursor home)
NUMENTER,C−P (sos doprefix)
	=
	[enter]
	[tabpre]
		set/extract number
		set/extract pending
		set/extract prefix
		set synonym
			sos doprefix
			sos prefix
			sos tabfieldf
			sos tabfieldb
				inprefix()
				pending.1()
				pending.2()
				pending.3()
				pending.4()
				prefix.1()
				prefix.n()
					a (add)
					c, cc (copy)
					d, dd (delete)
					f (insert following)
					i (insert)
					m, mm (move)
					p (insert prior)
					s (all)
					x, xx (exclude)
					/ (make current)
					",""
					(duplicate)
					>,>> (shift right)
					<,<< (shift left)
					.a (bookmark)
					TABL
					SCALE

Back to topics list.

Macros

keys	command	set command	sos command	function	prefix

	cmsg
	editv
	emsg
	enter
	hit
	macro
	msg
	nomsg
	nop
	preserve
	readv
	refresh
	reset
	restore
	rexx
		set/extract clearerrorkey
		set/extract cmdarrows
		set/extract compat
		set/extract curline
		set/extract impmacro
		extract lastkey
		extract lastmsg
		extract lastrc
		set/extract macro
		set/extract macroext
		set/extract macropath
		set/extract msgline
		set/extract msgmode
		set/extract reprofile
		set/extract rexxoutput
				altkey()
				batch()
				ctrl()
				initial()
				lastmsg.1()
				lastrc.1()
				shift()	

Back to topics list.

Keys

keys	command	set command	sos command	function	prefix

SPACE	cmatch
	controlchar
	define
	enter
	hit
	readv
	reset
	showkey
	sos
		set/extract clearerrorkey
		set/extract cmdarrows
		set/extract compat
		extract lastkey
		set slk
		set/extract typeahead
		extract showkey
				altkey()
				ctrl()
				shift()
				showkey.1()
				showkey.n()	

Back to topics list.

Miscellaneous

keys	command	set command	sos command	function	prefix

		set/extract clearscreen
		set/extract etmode
		set/extract mouse
		set/extract nondisp
		set/extract xterminal
		extract length
		extract monitor
		extract terminal
		extract version
				blank()
				current()
				spacechar()
				length.1()
				monitor.1()
				monitor.2()
				terminal.1()
				version.1()
				version.2()
				version.4()	

Back to topics list.

Last changed on: 31 Oct 1998 by Franz−Josef Wirtz<fw@ecotopia.oche.de>

Return to Table of Contents

APPENDIX 7 − REGULAR EXPRESSIONS IN THE

This appendix contains details on regular expression usage in THE. There are two places where THE uses regular expressions; in targets in commands like LOCATE and ALL , and in the specification of patterns in THE Language Definition files used for syntax highlighting.

THE uses the GNU Regular Expression Library to implement regular expressions. This library has several different regular expression syntaxes that can be used when specifying targets.

Note that all pattern specifications used for syntax highlighting always uses the EMACS regular expression syntax.

The following table lists the features of each of the regular expression syntaxes that can be set via the SET REGEXP command. Each feature in the table is explained later.

This appendix is not intended to explain everything about regular expressions. If you want to find out more about GNU Regular Expressions, then view the on−line documentation at http://www.hessling−editor.sf.net/doc/regex/regex/ .

Syntax	Features
EMACS	None set
AWK	BACKSLASH_ESCAPE_IN_LISTS DOT_NOT_NULL NO_BACKSLASH_PARENS NO_BACKSLASH_REFS NO_BACKSLASH_VBAR NO_EMPTY_RANGES UNMATCHED_RIGHT_PAREND_ORD
POSIX_AWK	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES CONTEXT_INDEP_ANCHORS CONTEXT_INDEP_OPS NO_BACKSLASH_BRACES NO_BACKSLASH_PARENS NO_BACKSLASH_VBAR UNMATCHED_RIGHT_PAREN_ORD BACKSLASH_ESCAPE_IN_LISTS
GREP	BACKSLASH_PLUS_QM CHAR_CLASSES HAT_LISTS_NOT_NEWLINE INTERVALS NEWLINE_ALT

EGREP	CHAR_CLASSES HAT_LISTS_NOT_NEWLINE NEWLINE_ALT CONTEXT_INDEP_ANCHORS CONTEXT_INDEP_OPS NO_BACKSLASH_PARENS NO_BACKSLASH_VBAR
POSIX_EGREP	CHAR_CLASSES HAT_LISTS_NOT_NEWLINE NEWLINE_ALT CONTEXT_INDEP_ANCHORS CONTEXT_INDEP_OPS NO_BACKSLASH_PARENS NO_BACKSLASH_VBAR NO_BACKSLASH_BRACES INTERVALS
SED	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES BACKSLASH_PLUS_QM
POSIX_BASIC	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES BACKSLASH_PLUS_QM
POSIX_MINIMAL_BASIC	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES LIMITED_OPS
POSIX_EXTENDED	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES CONTEXT_INDEP_ANCHORS CONTEXT_INDEP_OPS NO_BACKSLASH_BRACES NO_BACKSLASH_PARENS NO_BACKSLASH_VBAR UNMATCHED_RIGHT_PAREN_ORD
POSIX_MINIMAL_EXTENDED	CHAR_CLASSES DOT_NEWLINE DOT_NOT_NULL INTERVALS NO_EMPTY_RANGES CONTEXT_INDEP_ANCHORS CONTEXT_INVALID_OPS

	NO_BACKSLASH_BRACES NO_BACKSLASH_PARENS NO_BACKSLASH_REFS NO_BACKSLASH_VBAR UNMATCHED_RIGHT_PAREN_ORD

BACKSLASH_ESCAPE_IN_LISTS

If this feature is not set, then \ inside a bracket expression is literal.
If set, then such a \ quotes the following character.

BACKSLASH_PLUS_QM

If this feature is not set, then + and ? are operators, and \+ and \? are literals.
If set, then \+ and \? are operators and + and ? are literals.

CHAR_CLASSES

If this feature is set, then character classes are supported. They are:
[:alpha:], [:upper:], [:lower:], [:digit:], [:alnum:], [:xdigit:], [:space:], [:print:], [:punct:], [:graph:], and [:cntrl:].
If not set, then character classes are not supported.

CONTEXT_INDEP_ANCHORS

If this feature is set, then ^ and $ are always anchors (outside bracket expressions, of course).
If this feature is not set, then it depends:
^ is an anchor if it is at the beginning of a regular expression or after an open−group or an alternation operator;
$ is an anchor if it is at the end of a regular expression, or before a close−group or an alternation operator.

This feature could be (re)combined with CONTEXT_INDEP_OPS, because POSIX draft 11.2 says that * etc. in leading positions is undefined.

CONTEXT_INDEP_OPS

If this feature is set, then special characters are always special regardless of where they are in the pattern.
If this feature is not set, then special characters are special only in some contexts; otherwise they are ordinary.
Specifically, * + ? and intervals are only special when not after the beginning, open−group, or alternation operator.

CONTEXT_INVALID_OPS

If this feature is set, then *, +, ?, and { cannot be first in an RE or immediately after an alternation or begin−group operator.

DOT_NEWLINE

If this feature is set, then . matches newline. If not set, then it does not.

DOT_NOT_NULL

If this feature is set, then . does not match NUL. If not set, then it does.

HAT_LISTS_NOT_NEWLINE

If this feature is set, nonmatching lists [^...] do not match newline. If not set, they do.

INTERVALS

If this feature is set, either \{...\} or {...} defines an interval, depending on NO_BACKSLASH_BRACES.
If not set, \{, \}, {, and } are literals.

LIMITED_OPS

If this feature is set, +, ? and | are not recognized as operators. If not set, they are.

NEWLINE_ALT

If this feature is set, newline is an alternation operator. If not set, newline is literal.

NO_BACKSLASH_BRACES

If this feature is set, then `{...}' defines an interval, and \{ and \} are literals. If not set, then `\{...\}' defines an interval.

NO_BACKSLASH_PARENS

If this feature is set, (...) defines a group, and \(and \) are literals. If not set, \(...\) defines a group, and (and) are literals.

NO_BACKSLASH_REFS

If this feature is set, then \ matches . If not set, then \ is a back−reference.

NO_BACKSLASH_VBAR

If this feature is set, then | is an alternation operator, and \| is literal. If not set, then \| is an alternation operator, and | is literal.

NO_EMPTY_RANGES

If this feature is set, then an ending range point collating higher than the starting range point, as in [z−a], is invalid.
If not set, then when ending range point collates higher than the starting range point, the range is ignored.

UNMATCHED_RIGHT_PAREN_ORD

If this feature is set, then an unmatched) is ordinary. If not set, then an unmatched) is invalid.

Return to Table of Contents

History of THE

Version 3.1 13−Aug−02

Bug fixes:

- Fix bug with QUERY ECOLOUR on Win32 platform.
- Entering a shifted character in response to FILLBOX command on X11 port resulted in incorrect values being inserted.
- Fixed compilation errors on platforms that don't have KEY_MOUSE in curses.h
- Fixed compile error on AIX when using REXX/6000.
- Fixed bug with SET MARGINS; second parameter did not allow '*' to be specified.
- Fixed bug with ZONE command when only the first parameter was supplied.
- Fixed small bug with identification of magic numbers in new files when the first line was copied into the file.
- Changed limit on number of files that can be edited from 256 to 2,147,483,647
- Fixed bug with sorting a file with only 1 line.
- When two different files are displayed as a result of SCREEN command, and COPY BLOCK (box block) is performed, block in source window does not get reset. Now fixed.
- Changed HTML TLD comment terminator from "−−>" to ">"
- Fixed bug introduced in 3.0; QUERY and EXTRACT would not recognise abbreviated keywords. eg. QUERY ARB would fail and you needed to use QUERY ARBCHAR.
- When editing a symbolic link, unless BACKUP INPLACE is ON, the symbolic link was lost. This is now fixed.
- Display of pending prefix indicator in STATAREA now clear.
- Added workaround for bug in AIX curses, where the cursor would move down one line when typing text into the last displayed column and the screen autoscrolled.
- Fix for column position being calculated as a negative value on some platforms, resulting in scrolling to the left beyond column 1.
- X11 port of THE more stable on startup (with PDCurses 2.5)
- Fix bug with changing file name, path or extension when run in batch mode.
- Fix bug with resizing of THE window to a small number of columns; still potential crashes if the window is resized to < 10 columns.
- Functions: focuseof() and focustof() returned the wrong value when on the command line and the curos was last on a line other that 'top−of−file' or 'bottom−of−file'.
- Fix error with wrong command being repeated (with REPEAT) command in a macro.
- Minor changes to deregister dynamic functions like parser.1() so that Object Rexx 2.2 on Linux does not leak shared memory and cause THE to crash.
- Fix bug with syntax highlighting for LABELS.
- Comments delimiters defined in TLD files can now be specified as case−insensitive.
- Fixed bug with GET CLIP: under Win32. Now doesn't crash after multiple uses of GET CLIP:
- Use of ALT keyword on :function definitions in a TLD file did not work.
- Remove limit on 20 directories in MACROPATH; now no effective limit.
- Fixed Rexx/Trans DLL on Win32 platforms that caused THE to not work with Object Rexx on Win2K and XP.
- Allow CMATCH command to be executed from a batch command.
- Disallow CURSOR HOME from a batch command.

- Allow C−C, C−S and similar commands in OS/2 port (fix to PDCurses)
- Fix bug with SET TRAILING and empty file.
- Allow more flexible specification of hex and decimal strings in targets. / x'3e2d' / is now identified as a valid hex string target (with HEX ON of course). Bug #531800

Changed commands or behaviour:

- Any ECOLOUR character can be used in the 'alternate' colour specifier for 'keywords' in a TLD definition. Previously, only the 'alternate' characters 1 through 9 were allowed.
- Added optional targets on EXTRACT PENDING command.
- Color on X11 port now behaves differently with PDCurses 2.5 and above. "Bold" colors are now drawn in the same font, but in a brighter color. This means that any monospaced font can be used; there is no need to have an equivalent bold font.
- "Blink" color "modifiers" on X11 port now displays background in bright color.
- Added extra color "modifier"; italic on X11 port. To display text in italics, you must have a normal font and italic font of the same size.
- Cursor on X11 port can now be set to blink with PDCurses 2.5 and above.
- The "find" family of commands can now be run without the mandatory string parameter. Like LOCATE, these commands can use the string used in the last "find" command if no parameter specified.
- [SET] COLOUR and [SET] ECOLOUR have options to turn on or off display modifiers like BOLD, BLINK etc. These can be set for individual colour specifications or for all colour specifications.
- Profile files specified with the −p command line switch will use the same mechanism for locating the file as is done with the THE MACRO command. ie THE_MACRO_PATH=/usr/local/THE set and mymacro.the is in /usr/local/THE; the −p mymacro will find /usr/local/THE/mymacro.the
- When setting variables via EXTRACT with Regina, THE now uses direct setting rather than symbolic setting of variables. Should be no user effect.
- POPUP command now returns POPUP.2 which contains the item number selected or 0 if nothing selected and POPUP.3 which is the item number of the line last highlighted.
- POPUP command now has no effective limit on what can be poped up. If the number of lines or columns is too large for the size of the screen, scrolling is now enabled.
- SET FULLFNAME now switches between displaying the fully qualified filename of the current file and the FNAME omponent of the file.
- Rewrote build.the macro to use POPUP command and to not stay in the macro after the build is executed.
- Number of lines in SET MSGLINE can be specified as * to indicate as many lines as will fit on the screen. The number of lines argument is now validated against the screen size and starting line argument. A new CLEAR argument is also provided to clear the msgline contents.
- Slight change with syntax highlighting; brackets are not coloured if there is no MATCH heading in the associated TLD file or if SET HEADER MATCH OFF is set.
- New optional specification for coloring functions not specified explicitely.
- Minor changes to syntax highlighting of numbers.
- SCHANGE now executes the previous arguments when no arguments are specified.

New features:

- Support for GNU Regular Expressions in LOCATE and ALL commands. Command syntax is LOCATE Regexp /RE/ and ALL Regexp /RE/ respectively.
- Added new SET REGEXP command to allow specification of which Regular Expression syntax that is to be used in targets.
- Added new Appendix (7) for explaining some details of Regular Expressions.
- New command line option; "1" for X11 port. This runs THE in "single window mode". What this means is that every time THE is started, it checks to see if THE is already in "single window mode" for the curent user, with the same optional filename supplied. If so, the file or files requested to be

edited on the command line are sent to the running instance and are added to that instances ring. Values from command line of −l, −c and −r passed to server and applied to the file being edited. Works on Unix with XCurses and ncurses.

- Added SYNONYM command. Not a full implementation yet; doesn't support parameter reordering.
- Add support for Rexx/Trans on Unix and OS/2 platforms.
- Added new command SET EQUIVCHAR to set a character to be used in parameters that would normally take '=' to specify a value from the current file or view. Default value is '='.
- All SET commands can now use EQUIVCHAR where appropriate.
- Any command parameter that specifies a filename can use the EQUIVCHAR to specify portions of the filename.
- Ports of THE now available for Amiga, BeOS, AtheOS, Linux/390 and Mac OSX.
- Added QUERY/EXTRACT NBSCOPE; number of lines in scope.
- Added EXTRACT FIELDWORD to return the current word in the current field.
- Added new command, REDIT, to discard the changes to the current file and reload it back into the ring retaining the current location.
- Added BOUNDMARK, a new SET command, and associated QUERY/EXTRACT and COLOR setting. BOUNDMARK displays vertical lines between characters indicating columnar areas like ZONE or TABS. Only significant on X11 port of THE with PDCurses 2.5 or above.
- Implementation of :number, :column and :postcompare headers in a TLD file.
- Added initial COBOL, MAKE and OPL TLD files in the distribution.
- New command; SET HEADER, to turn on or off portions of the current syntax highlighting.
- Added optional Common User Access (CUA) commands to allow THE to behave in a manner consistent with the CUA definition. New commands; SOS CUADELBACK, SOS CUADELCHAR. Changed commands; CURSOR now has a CUA option for UP, DOWN, LEFT, and RIGHT. enter command now has optional CUA option. MARK command also has a CUA option to mark a CUA block, which is like a STREAM block, but acts like the CUA mark. See the sample cua.the macro for setting up THE to work like a CUA editor.
- Added new config.the macro. It is a GUI configuration tool for THE but is incomplete at this stage. This requires Rexx/Tk 1.2 or above to work.
- Added new sample macro; tags.the. This is intended to be used with Exuberant Ctags.
- Added new sample macro; nl.tld. This displays a POPUP with all the named lines in the file and moves the cursor to the selected line.
- Added new sample macro; complete.the. This macro provides code completion for Rexx and C languages. Should be easy to extend it for other languages.
- Added new colour setting capabilities; CPREFIX and CGAP. These are equivalent to PREFIX and GAP respectively, but relate to the current line.
- Added THIGHLIGHT support; RESET, SET/MOD/QUERY/EXTRACT THIGHLIGHT, and [SET] COLOUR THIGHLIGHT. THIGHLIGHT displays found string or RE targets in the colour specified by [SET] COLOUR THIGHLIGHT. Also string targets in SCHANGE command also displayed in thighlight colour.
- Added [SET] TRAILING to change behaviour of trailing blanks in files.
- Added preliminary compatibility mode for ISPF. Changes include SET COMPAT, display of "TABS>", "COLS>" in prefix area for tabs and cols indicator lines. The changes are incomplete.
- Added new command; CURSOR GOTO. This is similar to CURSOR FILE, but will move the cursor the the specified line/column even if they are not in view.
- Added new SET command; TARGETSAVE. This allows you to specify which target types are saved for subsequent calls to the LOCATE command without any parameters. By default; SET TARGETTYPE ALL, the LOCATE command without any parameters, locates the last target irrespective of the type of target. This is the current behaviour. With SET TARGETTYPE STRING then the only target saved will be one that has a string target component. ie. if you executed LOCATE /fred/ then LOCATE :3 then LOCATE, the final LOCATE will look for /fred/ NOT :3
- Added new SET command; PAGEWRAP. This allows the user to specify if FORWARD and BACKWARD commands wrap to the bottom and top of files automatically or not.
- Added new SOS command; SOS CURSORSHIFT. Similar to SOS CURSORADJ but shifts text to

the right of the cursor to the cursor position.

- Added QUERY/EXTRACT EFILEID to return the original filename of the current file.
- Added QUERY/EXTRACT LASTOP to return the last operand specified for various commands.
- Added SET FILETABS, TABFILE command and SET COLOUR/COLOR FILETABS and FILETABSDIV. FILETABS allows for a single−line window displayed at the top of the screen containing the filenames of all files (except the current file) in the ring. Clicking the mouse on a filename makes it the current file.
- Added new prefix commands for all compatibility modes:

```
LC − lowercase line(s)
UC − uppercase line(s)
LCC − lowercase block
UCC − uppercase block
( − shift left column(s) within zone
) − shift left column(s) within zone
(( − shift left column block within zone
)) − shift left column block within zone
Not implemented yet...
O − overlay line(s) − same as OVERLAYBOX (except ISPF mode)
OO − overlay block(s) − same as OVERLAYBOX (except ISPF mode)
```

- Added or modified prefix commands for ISPF compatibility mode:

```
A − after target
B − before target
R − repeat line(s)
RR − repeat block
O − overlay line(s) − (ISPF behaviour)
OO − overlay block(s) − (ISPF behaviour)
COLS − column indicator line
TABS − tabs indicator line
BOUNDS − bounds indicator line
```

Version 3.0 16−Jan−00

Bug fixes:

- Fix bug with QUERY RING; not all files were displayed depending on MSGLINE setting.
- Implied extract functions for RING now work for all files in the ring. Thus if you have 20 files in the ring ring.20() will return the details about the 20th file in the ring. If you close one file ring.20() will result in an error.
- Fixed some formatting errors in the generated HTML documentation.
- Allow 'C' as mouse button action (click) in DEFINE command. eg DEFINE CLB in FILEAREA... (click left button). Not all curses libraries support this.
- Fixed bug in COPY BLOCK of line block from one file to another. Sometimes after the copy, any CHANGE command with BLOCK as target would fail.

Changed commands or behaviour:

- [SET] MACROPATH can now take the string PATH as an option. This sets the search path for macro files to be the system PATH as specified in the PATH environment variable.
- When using PDCurses and ncurses, the colours "grey"/"gray" and "white" have been swapped. Thus white is now brighter than grey.
- Remove restriction on the use of '/' as the only delimiter character for EXTRACT command.
- Fixed bug in locating a string when SET WRAP ON and the focus line has had changes made to it.

New features:

- Syntax highlighting! Added support for syntax highlighting, modeled on KEDIT for Windows syntax highlighting. Commands added: [SET] COLOURING/COLORING, [SET] AUTOCOLOUR/AUTOCOLOR, [SET] PARSER, [SET] ECOLOUR/ECOLOR. New appendix in documentation to explain how syntax highlighting works and the format of language definition files. Also added AUTOCOLOR, AUTOCOLOUR, COLORING, COLOURING, ECOLOR, ECOLOUR, and PARSER to QUERY/EXTRACT commands.
- Added [SET] CTLCHAR and support in [SET] RESERVED to allow reserved lines to be displayed in different colours.
- Added new command line switch; −q, to suppress the message that is displayed when error messages are displayed from execution of the profile file.
- Started a FAQ.
- Added the following valid key names; SHIFT−L, SHIFT−R, CONTROL−L, CONTROL−R, ALT−L, ALT−R. These correspond to pressing the equivalent modifier keys by themselves. The CONTROL−R should be useful for XEDIT users! Note that not all platforms support the return of key events when pressing the modifier keys by themselves. Also be careful with assigning commands to the modifier keys and then using the modifier keys to modify other keys; as some curses implementations will ALWAYS return a modifier key event followed by the "normal", modified key press.
- Added [SET] LINEFLAG command to change the characteristics of a line.
- Added POPUP, DIALOG and ALERT commands.
- Added extra [SET] COLOR options for ALERT and DIALOG commands.
- When used with PDCurses 2.4 (non−beta version), the X11 port of THE has a new switch command line; −X. This allows the specification of both standard X11 switches, such as "−iconic", "−display" as well as XCurses−specific switches such as "−colorBlack" which correspond to the XCurses resources that can be set via .Xdefaults file.
- Added optional specification of soft label key format.
- Added SET COMPAT KEDITW (Kedit for Windows) and default colours.

Version 2.8 31−Jul−99

Bug fixes:

- Fixed a bug with THE functions returning a value > 255 characters.
- THE incorrectly executed the profile file against a file edited from within the profile file.
- EXTRACT /GETENV would always return an invalid value.
- If the current directory were changed from within the edit session, the new directory was not being used when determining the path when editing a file.
- EDITV LIST now shows full variable name.
- THE now registers its external functions in both upper and lower case. This is done to cater for Rexx interpreters that are sensitive about case with external functions; noteably REXX/imc (Patch by Ian Collier)
- Files edited on AFS mounted file systems under HP−UX should now work.
- Fixed bug with SPLIT and JOIN commands when run from the command line; the current line was advanced incorrectly.
- Fixed bug with "ls ~"; now returns correct directory listing.
- SOS DELLINE will now save the line being changed before it deletes it. This is done so that if you happen to be typing the line and delete it before moving off the line, RECOVER will get it back correctly.
- The first() boolean function was defined incorrectly. It was essentially defined to be the same as leftedge(). It now correctly reports 0 if on the left edge of the FILEAREA, but VERSHIFT is > 0.

- CURSOR CMDLINE col; now positions the cursor at the specified column correctly, when the command is run from the command line or from a macro run from the command line.
- STREAM blocks support added for COPY and MOVE. Still to do: PUT.
- When autosaving files as a result of a trapped signal, turn off the signal handler, and only save files that have outstanding changes.
- DEL −* when the current line is more than half way down the file used to leave one line undeleted which should have been deleted.
- COPY −* to a target now works correctly.
- Don't execute the profile file against a file when that file is already in the ring.
- Fixed bug with allowing pseudo files to be saved.
- Fixed the error that indicates that line 0 is longer than the maximum line length. Now correctly indicates line 1.
- Fixed crash when MOVE BLOCK is execute when there is no marked block.

Changed commands or behaviour:

- Added new default mouse button behaviour; DEFINE A−PLB in FILEAREA BACKWARD DEFINE A−PRB in FILEAREA FORWARD
- OSREDIR now returns the error from the command executed.
- READV KEY now recognises mouse events.
- HIT also now recognises mouse key definitions.
- The last target value used by the LOCATE command without any parameters, is not changed by LOCATE commands run from within a macro.
- PUT command now valid in readonly mode.
- In XEDIT compatibility mode, reserved lines are not copied to a file when that file is brought into the ring.
- Lines with a pending prefix command are now not displayed as a result the ALL command unless the line contents match the target in the ALL command.
- Allow XX,X and S prefix commands in readonly file.

New features:

- Added a new SET command; READONLY. This has the same effect as the −r command line switch, but only affects those files that were read into THE as readonly files. Option FORCE forces ANY file to be readonly.
- Added SET TOFEOF command, to turn on or off the display of the top−of−file and bottom−of−file lines.
- Added new macro; build.the. This is a more sophisticated attempt at an IDE in THE than compile.the. See also setbuild.the.
- The pseudo−file CLIP: can be used in place of the filename argument in the PUT and GET commands. This functionality available on OS/2, Win32 and X11 ports of THE with version 2.4 and above of PDCurses. "CLIP:" is the system clipboard or equivalent (selection under X11).
- Added INPLACE option to SET BACKUP. This option preserves all system file attributes of the edited file. The original file is emptied BEFORE the new contents are written. Thus there is a chance that you will lose your file, but the file retains all of its original attributes.
- Added mouse support for ncurses−based ports of THE. Note that ncurses does not support mouse movement, so dragging the mouse to mark a line or box block does not work.
- Ports of THE using ncurses now resize correctly if the terminal resizes and issues a SIGWINCH interrupt. QNX pterm works fine, but xterm and nxterm need an extra keystroke to refresh the screen correctly. (Only tested with ncurses 4.2). (I believe this is a "feature" of ncurses).
- Added TAG command to highlight all lines that contain a target. Similar to ALL, but all lines are displayed and tagged lines are shown in a different colour.
- Added optional line/column specifiers to MARK command allowing the user to explicitly specify the dimensions of a marked block.

- Added new QUERY option; REXX to return version details of Rexx interpreter being used by THE. Output is identical to Rexx command PARSE VERSION.
- THE now works on QNX.

Version 2.7 27−Dec−98

Bug fixes:

- On some systems, THE would crash after the first several keystrokes.
- On some systems, blank lines would match every string target.
- Printing blank lines sometimes produce junk.
- Version information on WinNT was incorrect, as was some documentation files.
- Use of OSREDIR command in profile file when suspended on Unix would cause THE to stop.
- Compilation errors with non−ANSI compilers fixed.
- Included extra HTML documentation courtesy of Franz−Josef Wirtz.
- Re−released 16bit Windows THEdit shareware package.

Version 2.6 14−Nov−98

Bug fixes:

- The screen would not be refreshed as a result of issuing Rexx commands via THE's REXX command. It now does.
- The implied extract functions; colour.n() (not the color.n() functions) were failing.
- SLK display line was not being refreshed when it was overwritten.
- SLK OFF was unsupported despite the documentation saying it was.
- EXTRACT FMODE would return "1" instead of the drive letter.
- The first row of the message line was not being cleared after a QUERY command on some platforms.
- THE would not compile on AIX 4.2 with "standard" curses, due to some changes in IBM's curses code.
- Under some Unix platforms, THE would crash on exit when an error or warning message was displayed.
- Return code from SET ETMODE ON|OFF with no following parameters now returns correct value; 0.
- Column commands such as CINSERT, CAPPEND would leave a nul character, 0x00 at the end of the line.
- The fix in 2.5.2 for editing a file from the DIR.DIR file, when the file is a symbolic link, didn't work when the symbolic link was an absolute path.
- Fixed MARK STREAM command. STREAM blocks now supported for following commands: LOWER, UPPER, FILLBOX, DELETE, CHANGE. Still to do, COPY and MOVE.
- inblock() function now works for STREAM blocks.
- SOS TABWORDB and SOS TABWORDF now respect SET WORD ALPHANUM.
- Printing under OS/2 should now work.
- THE now configures correctly on HP−UX 11.0.

Changed commands or behaviour:

- When UNTAA is ON, the default value for certain commands like DELETE, REPEAT etc. is 1. This resulted in unexpected and undesirable behaviour. Commands that have an explicit default of 1, now have an explicit default of +1, to overcome the UNTAA behaviour. Thanks to Arthur Poole for pointing this out.
- A small performance increase should be expected for Rexx commands assigned to function keys,

using the DEFINE key REXX commands... In previous versions, THE would pass the source specified as an "instore" macro and ignore the returned, tokenised code. THE now saves the tokenised code and passes this to the Rexx interpreter. The first time the key is pressed may in fact be slightly slower, but subsequent executions should be faster. How much of a performance increase you get depends on how efficient your Rexx interpreter's parsing abilities are.

- Extended curses support now only allowed for AIX 3.x. As AIX 4.1 and above now has System V curses, this is the prefered curses library for the character mode version of THE.
- The Rexx indicator of the STATUSAREA, which used to be a space or 'R' to indicate if Rexx support was enabled has been changed slightly. When Rexx support is enabled, instead of 'R' being displayed, one of the following characters is displayed to indicate which Rexx interpreter support THE was compiled with. Values are:

```
R − Regina
O − Object or OS/2 Rexx
Q − Quercus Personal Rexx
W − Enterprise WinRexx
I − Rexx/imc
U − uni−REXX
6 − REXX/6000
T − Rexx/Trans (Rexx Translation package)
```

- FILLBOX now allowed for LINE BLOCKS.
- What constitues a word with the MARK WORD and DELETE WORD commands has now changed. It now matches the SOS TABWORDF definition of a word.
- READV now sets 4 Rexx variables, the new fourth value is the shift status associated with the key being pressed.
- A new optional parameter has been added to the MACRO command. Specifying this; a '?', enables Rexx macros to be interactively traced with TRACE ? in the macro file. In previous versions, this was the default for MACRO, but it involved a significant processing overhead. Now, Rexx command files assigned to keys should execute quicker.

New features:

- Added new SET command; CLEARERRORKEY. This enables you to specify which particular key clears the message line. Default behaviour is that any key pressed will clear the message line. Also added CLEARERRORKEY to query/extract/modify.
- Added query/extract/modify for DIRINCLUDE.
- Added new SET command; TIMECHECK. This enables you to specify if THE checks the timestamp of a file when you SAVE or FILE it. If TIMECHECK is ON and the modification time of the file has changed, THE will not save the file. You can force the save of the file with SSAVE or FFILE. Also added TIMECHECK to query/extract/modify.
- The PRINT command in Win32 now can specify fonts, font size, paper orientation etc.
- New command line options; −l and −c. These specify the starting current line and column respectively for all files edited from the OS command line.
- A new SET command has been included; STATOPT. This controls what internal variables are displayed on the status line. Any value that can be obtained via the implied extract functionality can be displayed. eg the default options displayed are NBFILE.1 and WIDTH.1.
- Added new boolean functions ALT, ALTKEY, CTRL and SHIFT.
- Added EXTRACT LASTKEY
- Added KEDIT command.
- THE now allows stdin as a profile file. When specifying the profile file on the OS command line with −p, the filename is '−'. Thus; the −p − file_to_edit takes macro commands from stdin. This feature is only allowed when the −b (run in batch) switch is also specified!! This feature is useful for porting XEDIT/REXX programs like:

```
/* CMS */
queue 'top'
queue 'c/this/that/ all'
queue 'file'
'X THIS FILE A'
```

The equivalent for THE with Regina is:

```
/* THE/Regina */
queue 'top'
queue 'c/this/that/ all'
queue 'file'
'LIFO> the −b −p − this.file'
```

Version 2.5.2 31−Jul−98

Bug fixes:

- Fixed problem with unaligned access problems on 64bit machines.
- Also fixed a few compiler warnings.
- Editing a file from the DIR.DIR file, when the file is a symbolic link, now correctly edits the file pointed to by the symbolic link.

Changed commands or behaviour:

- STATUS command now displays in 6 columns. On 25 row screens some status information was likely to be lost.

Version 2.5.1 28−Jul−98

Bug fixes:

- Fixed problem with compiling without Rexx support.
- Fixed the Win32 port mouse support. Mouse support would only work until a Rexx macro or an operating system command was called and then mouse support would be disabled.
- Fixed bug in COMPAT XEDIT mode where the behaviour of the "enter" command in the FILEAREA would behave like INPUT command rather than SOS ADDLINE/LINEADD.
- Fixed bug in display of cursor when in insert mode under several platforms.

Changed commands or behaviour:

- SOS ADDLINE/LINEADD under COMPAT XEDIT now works more like XEDIT.

Version 2.5 30−Jun−98

Bug fixes:

- Fixed problem with DIR or THE/XEDIT/EDIT command in a profile. THE would go into an infinite loop.
- LSCREEN values were not available as implied extract functions.
- Specifying 0 for row or column with CURSOR SCREEN/ESCREEN now causes an error.
- lastmsg.1 not being set when MSGMODE is OFF
- fixed bug with SET AUTOSAVE in profile

- fixed cursor positioning bug when an "edit" command was issued for a file already in the ring from a macro or key
- fixed error with alteration counts not being reset when saving a file under AFS.
- ALL command on a file with no lines caused THE to crash
- return codes from THE macros called by other macros should now be returned
- EXTRACT /INBLOCK/ now correct for all marked blocks (except STREAM)
- CINSERT will now insert spaces.
- implied extract functions, such as alt.1(), will fail if called when no files are in the ring.
- EXTRACT RESERVED did not include attribute modifiers when run on mono monitor.
- Fixed error with ACLs under HPUX where if the file system did not support ACLs, files could not be saved.
- Fixed error in PRESERVE/RESTORE where changes to the layout of the displayed screen were not being restored.
- Made changes to support API changes in Regina 0.08c.
- SET COMPAT would reset ALT settings.
- A command consisting of blanks returns an error; now simply ignored.
- With STAY OFF, PUT now moves the current line to the target line if in COMPAT XEDIT(feel) mode.
- EXTRACT /CURSOR with cursor in prefix area, now returns the file's line number in cursor.3.
- THE can now handle messages via MSG and EMSG commands > 160 characters.
- Fixed problem with dir() boolean function to return correct value.
- EXTRACT /RING in COMPAT XEDIT(feel) now returns the number of files in the ring in ring.1. For COMPAT THE and KEDIT the number of files is returned in in ring.0.
- With HEX ON, LOCATE /x'00' would match every line in the file.
- When an attempt to edit a file with a line width > the current WIDTH setting, THE would not close the file. On some platforms this meant that file was no longer accessible until the edit session was ended.
- Under Unix, THE now attempts to preserve the file's user and group ownership ids. This should work fine for the file's owner and for root, but probably won't for other users.
- Use of interactive trace in a THE macro would not work under OS/2, DOS or WIN32. Should now.
- Commands that are very long, particularly those issued from macros used to crash THE. Now they shouldn't.
- Fixed behaviour of CMATCH in COMPAT XEDIT mode.
- Fixed bug in COMPRESS command.
- Corrected SOS DELLINE and SOS ADDLINE behaviour when issued on cmdline.
- In COMPAT XEDIT, SET TABSIN caused the current line to be moved to the bottom of the file.
- THE now compiles on Digital Unix 4.0
- In COMPAT XEDIT mode, THE would allow characters from the prefix area to display in the gap between the prefix area and the filearea.
- THE now removes multiple, consecutive slashes from a filename. eg. /usr///include//stdio.h is now interpreted as /usr/include/stdio.h
- Fixed typo in INSTALL file −−with−−curseslibdir changed to −−with−curseslibdir
- Fixed an error with EXTRACT. If a space appears before the first item name to extract, an error was displayed and the last item was not extracted.
- Use of UPPER or LOWER commands did not set the LINEFLAG to changed.
- An invalid hexidecimal or decimal value in a string target now only results in one error message, rather than one per line of the file.
- The OS family of commands now return the return code from the operating system command.
- The definitions and behaviour of CURSOR SCREEN/ESCREEN row col were reversed.
- Fixed cursor positioning problems with SOS DELLINE when the *** Bottom of File *** line was the current line.
- Any commands that saved a file with a different name to the file being edited; eg PUT, SAVE and FILE, always kept the same attributes as the original. This has been changed so that any new file takes on default attributes (and ownerships) as would a new file.

- Several documentation corrections, particularly in the HTML source. Thanks to Franz−Josef Wirtz for reporting these.
- Fixed error with SET COLOR. The NONDISP colour would always get changed.
- Fixed a problem with unknown keys, such as decimal 226 (A−B), using ncurses. These now can be assigned commands with DEFINE (eg. DEFINE \226 TOP) and will display their correct assignment when using SHOWKEY.
- Trailing spaces on the command line will now be passed to macros.
- Fixed a bug on some System V R4 curses and ncurses, where the cursor would disappear after toggling the INSERTMODE to ON. This was caused by a lack of the terminal's capability to display the cursor in a high visibility mode, instead it would not display it at all.

New features:

- Added new SET commands; FILENAME, FNAME, FEXT, FTYPE, FMODE and FPATH.
- Added COLOR option to EXTRACT and QUERY commands.
- Added WIDTH display to status area.
- Added [SET] SLK to allow user to define soft label keys.
- Added SLK to [SET] COLOR.
- Added new command line option; −k to enable soft label keys support.
- HTML'd the HISTORY file.
- Added [SET] WRAP
- Added [SET] WORD
- Added [SET] WIDTH This enables the maximum width of a line to be set within an edit session.
- Added [SET] FULLFNAME to specify if the file's fully−qualified filename is displayed on the IDLINE or the name of the file as entered by the user is displayed. This command may change to include settings for whether the file's fully−qualified name is used when writing the file.
- On Unix platforms, if THE suffers a core dump, THE will attempt to autosave all files that are currently open. (Thanks to Ian Collier for this good suggestion)
- Added a new SET command; UNTAA. This is an ancronym for "Unsigned Numerical Targets Are Absolute". This SET command affects ALL numerical targets, not just the LOCATE command!
- Added SET COLOR/COLOUR GAP to enable the colour of the gap between the prefix area and the filearea to be specified. By default, COMPAT XEDIT will display the gap in the same colour as the filearea, COMPAT THE in the same colour as the prefix area.
- Added [SET] DEFSORT command to specify how files should be sorted in the DIR.DIR file. Also added DEFSORT as a valid option for EXTRACT.
- Added EDITV command for setting and retrieving persistent macro variables.
- Added SET SPAN, SPILL and TRUNC. They don't do anything yet.
- Added REXX command to allow Rexx instructions to be run from the command line.
- Included the Regina memory management routines into THE to increase its performance.
- Added SOS PASTECMDLINE SETTAB INSTAB STARTBLOCK and ENDBLOCK.
- Added EXTRACT FIELD.
- Added shadow() boolean function to determine if the cursor is on a shadow line.
- Under Win95/NT, THE can now use Object Rexx, Personal Rexx, Enterprise Rexx and uni−REXX interpreters in addition to the default interpreter, Regina.
- Added support for uni−REXX 2.7.0 under Unix.
- Added QUERY and EXTRACT COLOUR (alternate spelling for COLOR)
- Added [SET] UNDOING to enable the saving of changed lines to be avoided. This makes large changes or deletes slightly quicker at the expense of not being able to RECOVER those lines affected while UNDOING ON is in effect.

Changed commands or behaviour:

- EXTRACT RESERVED format has now changed. Originally the format of the returned string was:

```
line−number foreground background modifier text...
```

```
eg. −1 red white normal This is a reserved line
```

The new format of the returned string is: line−number modifier [modifier[...]] foreground on background text...

```
eg. −1 red on white This is a reserved line
    −2 bold reverse white on black This is a reserved line
```

This new format now ensures that the contents of the reserved line can be extracted from the returned value. The following example is guaranteed to work on colour or mono displays.

```
Parse Var reserved.1 . 'on' . text
```

- Added FULL as optional parameter to LEFT and RIGHT commands.
- Changed the behaviour of cursor movement in COMPAT XEDIT to be more consistent.
- Slight change in the way THE determines the name of a macro to execute. Assuming the value of MACROEXT is "the" and you have a macro called fred.the in the current directory, then the following commands are equivalent:

```
====> macro fred
====> macro fred.the
====> fred          /* assumes IMPMACRO is ON */
====> fred.the      /* assumes IMPMACRO is ON */
```

  In previous versions, THE would always append the MACROEXT, so macro fred.the would attempt to find the file fred.the.the.
- There is now only one help file; called THE_Help.txt rather than one for each platform/terminal type. THE_Help.txt replaces the *.hlp files. The only difference between these files was the default key definitions. These are now in the one table.
- Changed the comm.the and uncomm.the sample programs to support HTML comments.
- Changed the match.the sample program to support LaTex begin/end pairs.
- Added [Column|CURSOR] options to SPLIT and JOIN commands.
- Inserting of lines greater than WIDTH setting, either via INPUT or JOIN are now truncated, and the message "Truncated" displayed.
- Blank lines are now saved for use with RECOVER command.
- With the addition of the SET FNAME, FILENAME etc commands, it was necessary to change the value returned for EXTRACT /FNAME. Prior to this release, EXTRACT /FNAME returned the full filename of the current file. EXTRACT /FNAME in this release, returns the filename portion of the full file name; excluding the file's extension. To achieve the same functionality as before, use EXTRACT /FILENAME.
- Added a new make target; help. This builds the THE_Help.txt from the source code.
- QUERY command now ignores the current setting of MSGLINE and displays all lines without scrolling, provided they all fit on the screen!
- SET POINT did not allow numeric values. Now SET POINT .12 works as expected.
- Removed extraneaous screen displays in CANCEL, CCANCEL and SOS EDIT commands.
- File dates displayed in a DIR.DIR listing now have 4 digits. The result of this is that file names now start in column 38 instead of 36. If you have any macros that rely on this position, they will need to be changed.
- In previous versions, the setting of SET MSGMODE for a new view of a file would be obtained from the setting of the view from which the file was edited. Thus if MSGMODE was OFF in the current file, and another file was edited, then the value of MSGMODE for this file would also be OFF. This has now changed so that the the default setting of MSGMODE will be ON irrespective of the setting of MSGMODE in the current file. Obviously, if SET MSGMODE OFF is in your profile and SET REPROFILE is ON, then the new file will have a MSGMODE of OFF.
- Fiddled around with the colours GREY (and GRAY) and WHITE to try to get WHITE not be BOLD WHITE and GREY to be BOLD BLACK.
- Added OFF option to SET TABS to turn off all tab settings.

- The COMPAT command has changed slightly when the third parameter is not supplied (function key compatibility). Previously, if the parameter was not supplied, the default key mapping for the current compatibility mode would reset, removing any customised key definitions. The behaviour now is to leave any key definitions intact.

Version 2.4 01−Feb−97

Bug fixes:

- Fixed problem with key defined as SOS DOPREFIX EXECUTE and supplied l.the prefix macro.
- SPLIT command issued with cursor after the end of line now does not core dump.
- SOS EDIT on blank line now no longer core dumps.
- Fixed bug with "the −h" on OS/2.
- Fixed major problems with PRINT command; it ignored effect of SET SELECT. Also corrected printing of marked blocks to only print the contents of the block. ZONE settings are also noew respected for the PRINT command.
- Fixed various documentation bugs principally in EXTRACT section.
- SORT command did not respect SCOPE; now does.
- SHIFT command now respects a marked block.
- Fixed some problems with cursor positioning in prefix area when the prefix area includes a gap; as in COMPAT XEDIT.

New features:

- First release of port for DOS with VCPI. This executable also runs as an OS/2 program.

Version 2.3 17−Dec−96

Bug fixes:

- Fixed syntax error with [SET] ARBCHAR, now allows specification of ON or OFF without optional arbchar characters.
- Fixed cursor positioning errors with DUPLICATE and COPY BLOCK when new focus line would not be in currently displayed lines.
- Fixed multiple key definitions for the "DEL" key. Now each different key that could represent a "delete" key is called a different name; DEL, REMOVE, DC.
- Disable [SET] AUTOSAVE on psuedo files.
- Change commands with a parameter of '*' for number of occurrences to change, will use the file's line width to determine the maximum number of changes to make rather than 2147483001. The use of the rather large number made it appear that THE was stuck in an infinite loop.
- Fixed bug with HEX string targets.
- Fixed bug with segmentation fault caused by:

  ```
  ====> the file.one
  ====> statusline off
  ====> the file.two
  ```

  This fix also fixes the error where if you have set STATUSLINE OFF and then run SET COMPAT, which turns STATUSLINE ON.
- Fixed "feature" with exit codes from Unix commands called from the run_os() function. All exit codes were multiples of 512.
- Fixed bug with CDELETE command, positioning cursor incorrectly on line after deleting characters.

- Fixed bug with CAPPEND, CREPLACE and CINSERT which did not allow for leading or trailing spaces in text argument.
- Fixed bug with default file types displayed in DIR.DIR file. The documentation for [SET] DIRINCLUDE states that the default is *; ALL file types, but the actual default setting was NORMAL, ARCHIVE and DIRECTORY files. This caused me an hour of debugging when I couldn't see any files on a CD :−(
- Fixed bug with all SET commands that use the M (middle) position specifier. Commands like CURLINE, SCALE, TABLE etc. would core dump under Unix.
- Fixed minor bug with error messages displayed for invalid file name and/or file path.
- Fixed a problem with PUT command. If PUT was issued while in a read−only directory, the temporary file could not be created. The temporary file now is created in a known, writable directory.
- A bug with CLOCATE and an absolute target. If the cursor was located in the filearea and to the right of the current column position, the new current column position as specified by the absolute target would not be changed.
- Fixed a long−standing bug with block prefix commands. When one block prefix command is entered, and another prefix command (not the other end of the block prefix command), the block prefix command executes as though the other end of the block prefix command was entered.
- Fixed bug with [SET] CMDLINE OFF when run from profile file.
- Fixed bug with ETMODE default settings when running with XCurses.
- Fixed error message that is displayed when READV CMDLINE is called with no command line.
- Fixed implied extract functions; position.2() amd position.3(). They now produce the correct results.
- Fixed core dump when using GET command and file contains lines greater than maximum width.
- Worked around bug in ncurses; the cursor would disappear when in insert mode.
- Fixed cosmetic bug when marking a block in one view with an existing marked block in the other view in split screen mode.
- Fixed bug with X and XX prefix commands when current line was to be excluded.

New commands:

- Added PREVWINDOW command; complement of NEXTWINDOW; to move to the previous file in the ring.
- Added [SET] ALT command to allow the alteration counts to be changed.
- Added REPEAT command.
- Added [SET] MOUSE command and new options to QUERY, EXTRACT and MODIFY commands.
- Added [SET] HIGHLIGHT command and new options to QUERY, EXTRACT and MODIFY commands.
- Added PRESERVE and RESTORE commands.
- Added SET MACRO command.
- Added COMPRESS command.

New features:

- Multiple arbchar character matches now work in targets. Thanks to Regis Bossut for implementing this! ARBCHAR support in the CHANGE command still to be done :−(
- In mouse−supported ports, Right Button Press on the IDLINE will execute PREVWINDOW command. Also, double−clicking the Left Mouse Button on a file in the DIR.DIR file, will execute SOS EDIT.
- In memory REXX macros now supported with optional [REXX] keyword in DEFINE command. In REXX supported ports, you can now do something like:

  ```
  ====> define f1 rexx if after() then 'sos firstcol'; else 'sos endchar';
  ```

  This will then do the same as the SOS STARTENDCHAR command.
- Added Win32 platform support. THE now runs as a native, console application under Win95 and WinNT. The mouse is supported as are long file names.

- Added ability to reassign mouse events via the DEFINE command.
- Selectable highlighting of lines now available. It is possible to display selected lines in a diferent colour. See [SET] COLOUR for new HIGHLIGHT and CHIGHLIGHT options.
- Changed the file display in DIR.DIR under Unix platforms to show symbolic links as 'ls −l' shows them.
- Added support for changing the default behaviour of THE where the behaviour of XEDIT and KEDIT differ. eg. COVERLAY command.
- Added support for ACLs under AIX.
- High−order characters in the Latin−1 character set can now be entered with a "compose" key under XCurses port.
- THE error messages are written to the pseudo REXXOUTPUT file immediately after the command that caused the error.

Changed commands or behaviour:

- When multiple files are in the ring and you exit from one of these, the file in the ring PRIOR to the file exitted will become the new current file. The previous behaviour was to make the NEXT file in the ring the current file. The new behaviour is more intuitive, particularly when editing files from the DIR.DIR file and there are other files in the ring.
- QUERY RING has changed to report files in the ring beginning with the current file rather than the first file edited.
- Commands that can result in multiple lines being changed, like CHANGE, LOWER, UPPER, etc. will now update the alteration count once per command rather than once per line that has changed.
- Added support for the System V compiler under SunOS. If /usr/5bin is in the PATH before running configure THE, and you do not have gcc or acc, then the System V compiler (and curses) will be used.
- Allowed greater range of punctuation characters to be used as string target delimiters.
- Altered parsing of command line to allow for commands and arguments to abut one another. This allows for 'next 5' to be specified as 'n5' and also more obscure command syntax like 'upper$fred'; equivalent to existing 'upper /fred'. The downside of this is that macros which begin with a valid THE command abbreviation, and follow with a delimiter will be treated as a THE command, NOT as an implicit macro. eg. Suppose you have a macro called upp.fred, you have IMPMACRO ON, and you attempt to call that macro from the command line implicitly with ====> upp.fred, THE will attempt to execute the UPPER command with a target of the named line; .fred rather than your macro.
- Added support for multiple ?s in ? (retrieve) command. eg. specifying ??? will now return the third last command entered on the command line.
- The SOS ENDCHAR and SOS STARTENDCHAR commands now work while in the prefix area.
- CHANGE command, when the target is a BOX BLOCK, will now change strings within the column bounds of the BOX, rather than all occurrences on the line.
- DEFINE command now has option to assign functionality to mouse events. The syntax for this includes a specification of which THE window the mouse event is to be executed in. See the default mouse event assignations for valid window names. These names are a subset of the valid areas that can be changed colour with the [SET] COLOUR command.
- SHOW command now displays mouse event assignations.
- Added new subcommand to CURSOR; MOUSE. This command moves the text cursor to the last position that a mouse event occured. This command is intended for use when redefining mouse events.
- Added HIGHLIGHT and CHIGHLIGHT options to [SET] COLOUR command.
- Added LINEFLAG and INPUTMODE options to QUERY, EXTRACT and STATUS.
- Changed the behaviour of the ENTER key while in the FILEAREA. Originally, when INSERTMODE was ON, the ENTER key would add a blank line, and when INSERTMODE was OFF, the ENTER key would simply move down a line. The behaviour now is determined by the setting of INPUTMODE. With INPUTMODE OFF, the cursor always moves down a line. With INPUTMODE LINE, the default, a new line is always added when the ENTER key is pressed.

INSERTMODE FULL, which is intended to be be similar to XEDIT power input mode has not been implemented yet.

- The CHANGE command can now be run with no parameters, to re−execute the last change done.
- Added extra, optional parameters to [SET] COMPAT command. The first new parameter allows the user to change the feel of THE to mimic the default behaviour of XEDIT, KEDIT or THE. The second new parameter determines the default function key binding to be used, again compatible with THE, XEDIT or KEDIT. The first parameter of the [SET] COMPAT command determines how THE will look; the second how THE will feel, and the third the default function key bindings.
- Added extra parameters to [SET] PREFIX command to allow the user to specify the width of the prefix area and any gap between the prefix area and the filearea.
- With the addition of more punctuation characters as valid target delimiters (including the underscore character '_'), the CONTROL_CHAR command has been renamed to CONTROLCHAR.
- Several values returned by EXTRACT now respect XEDIT compatibility when in COMPAT = XEDIT. They are: CURLINE, LINE, LINEFLAG, POINT, SELECT
- OVERLAYBOX now supports line blocks.
- The A and I prefix commands now move the cursor to the file area.

Version 2.2 26−May−96

Bug fixes:

- Fixed a small memory leak when quiting from a file.
- Some colours were not behaving consistantly. This was corrected. eg REVERSE is now BLACK on WHITE, NOT reverse of the current colour. WHITE is now a bright white. The original WHITE is now GREY or GRAY.
- Fixed bug with SOS TABWORDB. When the first word in the line started in column 2, the cursor would go to column 1.
- Fixed bug with 'EXTRACT /POINT */'. This, or any parameter with arguments, would only work if it was the last parameter in the list of parameters to EXTRACT.
- Fixed small bug with MACROPATH.
- Fixed [SET] RESERVED to respect spaces after the last attribute specifier. Now it is possible to have spaces preceding the reserved text.
- under OS/2 (and probably DOS), when a DISK FULL error occured when writing a file, the file could not be closed and therefore could not be removed while returning the file to its original state.
- box commands COPY BLOCK and MOVE BLOCK now respect scope.
- Fixed bug with trailing spaces on command line. (Introduced in 2.1)
- Fixed bug under HP−UX with ACLs on NFS mounted disks. Files can now be saved, but any ACLs are lost on the original files.
- If CURLINE M was in effect, and MOD CURLINE was issued, the response would be SET CURLINE M+0. M+0 is an invalid specification.
- SET SCREEN now adjusts CURLINE setting when screen size changes.
- MODIFY command would always append a space after the SET command returned.
- Fixed problem with setting RESERVED, SCALE, TABL or HEXSHOW lines the same as CURLINE. You get an error if the position is the same as CURLINE. Similarly it is an error to set CURLINE to a position which is already occupied by one RESERVED, SCALE, TABL or HEXSHOW.
- Fixed bug with SET PREFIX OFF when issued from CMDLINE or from PREFIX area.
- Fixed bug with macros calling other macros. If an EXTRACT was issued from within one macro after calling another macro, the command would be rejected.
- Fixed bug with REXX macro containing REXX SAY command with no arguments and REXX output captured to a file; a line of junk would be inserted into the REXXOUTPUT file.
- Changed output of Q TABS to respond to INCR n if tabs were set with INCR option.
- Fixed problem with TABS output from 'STATUS file' to not truncate the tabs string.

- Fixed bug with contents of cmdline.3 after EXTRACT /CMDLINE/.
- Fixed bug with end() and blank() not returning correct values in certain circumstances.
- The retrieve last command command; ? retrieved the wrong command.
- Fixed core dump when attempting to edit a file in a directory that doesn't exist from the OS command line. eg

  ```
          % the xyx/abc
  ```

  where the path xyz does not exist
- Fixed core dump when issuing SET SCREEN with no operands.
- Fixed up CURSOR ESCREEN when issued in split screen mode. It wouldn't move to the correct place.
- On some platforms, a core dump would occur when trying to edit a file with lines longer than the maximum line length.
- Fixed bug with negative offset paragraph indents.
- Fixed bug with CURSOR FILE when display had been scrolled horizontally.
- Editing with split screens is more stable.
- PUT now respects SET STAY
- REDRAW command now redraws screen correctly.
- Fixed problems with moving WORD and COLUMN blocks.
- Fixed a bug that showed up in SET COMPAT where the redefinition of the ENTER key would cause a core dump.
- Fixed minor bug in parsing various [SET] commands. More than one space before a parameter would invalidate the parameter.

New commands:

- Added [SET] CURSORSTAY to set the behaviour of the cursor when scrolling the file with FORWARD or BACKWARD. Originally, with the cursor in the filearea, when a FORWARD or BACKWARD command was executed, the cursor would move to the current line (unless SET COMPAT XEDIT/KEDIT was in effect. The default behaviour is now to leave the cursor on the same screen line when the file scrolls (CURSORSTAY ON). This command allows for the original behaviour; with COMPAT THE, to be selected.
- Added new EXTRACT option; SHOWKEY. This option allows the macro writer to extract the commands assigned to a key after having run a READV KEY command.
- Added a new sample macro; spell.the. This provides spell checking capabilities in THE. This macro requires International Ispell Version 3.1 or above and REXX support.
- Added a new sample macro; demo.the. This macro is a self−running demonstration of THE concepts and commands. It requires REXX support. Run it via: the −p demo.the demo.txt
- Added FIND, FINDUP, FUP, NFIND, NFINDUP and NFUP commands.
- Added CLOCATE and CDELETE commands, but with no string targets.
- Added COLUMN option to QUERY and EXTRACT commands.
- Added BLOCK option to QUERY (already in EXTRACT) command.

New features:

- This release is the first to run as a native X11 application. Features unique to the X11 version include; resizing of the X window by dragging the window border and mouse support, including cursor positioning, marking blocks etc.
- The method of reading files from the command line and processing the profile file has changed dramatically. This was done to enable the processing of all commands from the profile file. As a result, a new comand line switch; −b, is required to execute a profile file against a file (or files) in batch mode. This is particularly necessary when run as a cron job under Unix.
- Added new external function; run_os(). This function enables macro writers the ability to run an OS command with stdin coming from a REXX "array", and stdout and stderr going to a REXX "array".

- Added [SET] XTERMINAL to specify which program to run when an OS, DOS or ! command is executed without arguments.
- The source distribution has been reconfigured with GNU autoconf. This should make configuring on different Unix platforms simpler.
- Added APPENDIX 2 to explain how THE handles the keyboard.
- Added APPENDIX 3 to document the default mouse behaviour.
- Added −u display_length command line switch to allow THE to be used as a non−line mode editor.
- A HTML version of THE reference manual can be generated from the source code. Run "make html" to build the HTML reference files. Start at index.htm.

Changed commands or behaviour:

- The name of the key associated with the curses KEY_RETURN has been changed from "ENTER" to "RETURN".
- Rationalised the naming standard for function keys. This could break existing profile key definitions. Check APPENDIX 2 for details.
- Added the optional argument; ALL to SHOWkey. This will create a new "pseudo" file showing all key default key definitions followed by any redefined keys. The format of the display is suitable for using as a macro to set all key defintions to the state they were in when the SHOWkey ALL command was issued.
- Default for MSGLINE is now ON 2 5 OVERLAY.
- Added * OFF option to [SET] RESERVED
- Added * option to 'EXTRACT /RESERVED */'
- Added TYPEAHEAD to output from STATUS, and allowed it as a valid QUERY and MODIFY option.
- Added CMDLINE option to READV command.
- Default setting for CURLINE in THE compatibility now M; same as for XEDIT and KEDIT compatibility modes.
- Added SCREEN to QUERY, MODIFY, and EXTRACT.
- Removed source files: extcurs.c, bsd.c
- When displaying 2 views of the one file, both views are updated simultaneously.
- Added RING option to QUERY and EXTRACT commands.
- Some default key definitions have changed; notably the default assignments for F6, F11, and F12 to be consistent across platforms and because '?' command now works :−)
- The CONTROL_CHAR and FILLBOX commands, will now leave the cursor positioned in the file rather than at the end of the message line.
- Changed the format of colour specifiers; THE now supports the optional "on" between foreground and background colours. eg: red on blue, is now allowed, as well as: red blue
- In XEDIT compatibility mode, it is now possible to move the cursor onto the *** Top of File *** and *** Bottom of File *** lines. This was done to enable the cursor to be moved with the mouse when the mouse is pressed above or below the file limits.
- The output from SHOWKEY ALL, now has commands enclosed in double quotes (") instead of single quotes ('). This was done to allow for key definitions that contain hex strings.
- Added NONE option to [SET] EOLOUT
- Expanded [SET] ETMODE ON to allow for specifying exactly which characters are to be displayed as themselves. This will be beneficial to non−english language users.
- Cleaned up the documentation and enabled generation of an HTML version of the THE manual.
- Allowed CMATCH command to be executed from command line, and more importantly, from within a macro.
- Added optional command to be executed after LOCATE command.
- Changed the key mapping significantly to provide the most logical key mapping for commonest keyboards; those with 12 function keys. See Appendix 2 for key mappings.
- SPLIT, SPLTJOIN and JOIN commands now work from the command line and act from the current focus column.

- More performance improvements have been made; LOCATEing a string with a trailing space is up to 7 times faster CHANGEing lines in a large file is also significantly faster.

Version 2.1 24−Jun−95

Bug fixes:

- Bug fixed with ALL followed by DEL *
- Could not find a named line if the named line was the "Bottom of File" marker.
- Fixed cursor positioning errors if [SET] SCALE, TABLINE or RESERVED were executed while in filearea and the cursor was on the line where the scale, tab or reserved line is to be displayed.
- copying lines from one file (with SELECT 1 − from ALL) copies lines NOT selected. SCOPE ALL also ignored in copy.
- PUT and PUTD did not copy the correct number of lines after ALL command
- QUERY TABKEY and SET TABKEY were not consistant. QUERY TABKEY returns settings as per SET TABKEY.
- OVERLAYBOX now respects SCOPE
- Fixed a core dump bug with implied extract under OSF/1.
- Scale line was not displaying past the end of ZONE.
- SOS DELWORD now positions the cursor correctly
- bug with 'c /x //' on some systems fixed.
- DUPLICATE, when assigned to a key, caused cursor to move strangely
- LEFT and RIGHT commands allowed in read−only mode.

New commands:

- Added OSREDIR command.
- Added a new sample macro; compile.the. This is a macro that will compile the current C program, and enable the stepping through, backwards and forwards, each error message, making each line associated with an error the focus line. Normal editing is still possible, although slower.
- Added CAPPEND,CFIRST,CINSERT,CLAST,COVERLAY,CREPLACE commands.
- Added COLUMN option to CURSOR command.
- Added LSCREEN option to EXTRACT command.

New features:

- HP−UX version will preserve Access Control Lists (acls) on files if the file is saved with the same name. The directory list (DIR.DIR) also indicates if a file has acls associated with it by displaying a '+' at the end of the file permission string.
- The OS/2 version preserves Extended Attributes on files if the file is saved with the same name.
- the 'pseudo' files DIR.DIR and REXX.$$$ are now REALLY pseudo files. No actual file is created as a result of the DIR/LS commands or from REXX output. Also the filename details displayed on the idline reflect the contents of the file.
- THE can now read Apple Macintosh text files.
- Although not strictly a new feature in this release, it should be mentioned that THE will work with both REXX/imc and REXX/6000 on each platform that each of these interpreters is available.

Changed commands or behaviour:

- SOS MAKECURR now correctly does nothing if executed from the command line
- Added another option to [SET] EOLOUT; CR, to write out files compatible with the Apple Macintosh.

- Several performance improvements have been made: Reading and writing of files is now up to 3 times quicker. Operations towards the end of files with many lines is now quicker.
- GET command now has the options, FROMLINE and NUMLINES to read a portion of a file.
- Default width reduced from 2048 to 512
- all commands should now allow trailing spaces on the command line
- Using BSD curses now incurs a penalty; the bottom line of the screen is not used. This is because BSD curses automatically scrolls the screen when a character is displayed in the bottom right corner. Now that THE can be configured to have different portions of the editor displaying on the bottom line, it was easier to reduce the number of lines rather than try to cater for all combinations of configurations.
- [SET] TAB can now use specific tab columns as well as INCR and QUERY TAB, and STATUS now display the actual tab columns in effect.

Version 2.0P1 11−Feb−95

Bug fixes:

- Fixed cursor positioning error when screen was scrolled to the left using CURSOR LEFT and SET VERIFY n m (where n > 1) was in effect.
- Changed MSG and CMSG so they work from within the profile file.
- Refresh the screen before accepting keystroke in READV.
- Allow minimum abbreviation for BOTTOM command to be B.
- Allow minimum abbreviation for [SET] SCALE command to be SCAL.
- Fixed problems when using COMPAT command in a profile.
- Fixed some problems with prefix macros.
- Fixed Extended Curses port so that a screen of other than 80x24 is now possible.

New commands:

- Added HIT command.
- Added FILE option to CURSOR command.
- Added a new sample macro; l.the. This is a prefix macro that can be used as a template for other prefix macros.

Changed commands or behaviour:

- Profile processing. Under Unix, THE now executes a "global" profile file from $THE_HOME_DIR and then the "local" profile file in $HOME.
- The order of Line, Col, Size on the IDLINE has changed in XEDIT compatibility mode to Size, Line, Col.

Version 2.0 26−Jan−95

Bug fixes:

- Fixed cursor positioning error when screen was an odd number of columns wide.
- Fixed problem with previous file's contents and command line remaining displayed when editting another file. (Only reported under AIX).
- Changed code again to handle multiple commands on the command line AND to ignore commands issued when no files are in the ring (excluding any 'edit' commands).
- Fixed DEFINE to not clear a key defintion until the validation of the new command(s) was complete.

- Fixed problem with core dumps when a command was passed to the operating system from within a profile file.
- Fixed bug in CURLINE when specfying a negative position. (Due to changing unsigned char to char)
- Fixed bug in displaying extended characters in ETMODE.
- Fixed bug in executing prefix commands after a TABPRE command.
- Fixed bug in SHIFT LEFT when the length of the line being shifted was < first column of ZONE.
- Text entered into the main window is displayed with the correct attributes.
- Fixed bug in entering prefix commands when in the last column of the prefix area; the cursor would wrap to the next line; now it stays in the last column of the prefix area.
- extracting values using an item abbreviation would result in the REXX macro being set to the abbreviation rather than the full name of the item. The full name of the extracted item is now used.
- fixed a bug when moving a box block.
- SPLIT, JOIN, and SPLTJOIN did not account for any pending prefix commands or marked block.
- Changed all references to keyboard return values from short to int. This is for support of DEC OSF/1 platform.
- The use of hex strings as an argument to the TEXT command defined to a key, caused problems. ie DEFINE F1 TEXT X'84' was altered after the first use of F1 key.

New commands:

- Added [SET] DISPLAY, SELECT, SHADOW, SCOPE.
- Added ALL command.
- Added CURSOR command.
- Added LEFT, RIGHT and RGTLEFT commands.
- Added CURSOR option to EXTRACT command.
- Added TABL,SCALE,X,XX,S prefix commands.
- Added [SET] RESERved , [SET] SCALE and [SET] TABLine commands.
- Added [SET] COMPAT command, to attempt to mimic the default behaviour of XEDIT and KEDIT.
- Added extra SOS commands:
  LEFTEDGE,RIGHTEDGE,PREFIX,QCMND,TABFIELDF,TABFIELDB,
  FIRSTCHAR,FIRSTCOL,LASTCOL,BOTTOMEDGE,TOPEDGE,CURRENT,
  MARGINL,MARGINR,PARINDENT,TABB.
- Added new macros, rm.the to delete from disk the file on the focus line of the DIR.DIR file, and words.the to count the number of words to a target.
- Added COLUMN and WORD options to MARK command.
- added READV to enable a REXX macro to obtain keystrokes from THE.
- added [SET] TYPEAHEAD, [SET] HEXSHOW commands
- added [SET] STATUSLINE, [SET] MSGLINE, [SET] IDLINE commands
- added [SET] POSITION command

Changed commands or behaviour:

- Default command for Alt−M/Ctrl−V is now MOVE BLOCK RESET not MOVE BLOCK.
- FORward 0 now makes the "Top of File" line the current line.
- BACkward 0 now makes the last line of the file the current line.
- When the "Bottom of File" line is the current line, FORward will make the "Top of File" line the current line. Similarly for BACkward when on "Top of File".
- [SET] CMDline OFF option added.
- [SET] ARBchar now supports multiple character matches
- In previous versions of THE, any line in a macro file, or profile file, that began with '/*' was treated as a comment line and the line ignored. For implementations of THE without REXX support, execution of a REXX macro caused mayhem. To overcome this situation, any macro file that is used with THE without REXX support must have as its first line the following comment string starting in column 1:

```
/*NOREXX*/
```

All other comments throughout the file can be valid REXX comments (ie. start and end with /* */)

- A new boolean function incommand() has been added which has the same functionality of the previous command(). The boolean command() function now returns 1 if the command line is on.
- Reinstated SOS EXECUTE command.
- the valid_target() external function now returns two values; the first line affected by the target and the number of lines affected by the target. If the target is an invalid target ERROR is returned. If the target is valid, but the target is not found, NOTFOUND is returned. Sample macros altered to use new features.
- NEXT and UP commands do not support negative relative targets. eg. NEXT −5 , UP −* are now invalid commands. This is to be consistent with XEDIT and KEDIT.
- ADD command now can be specified as just "A" and when issued from the command line, moves the cursor to the first blank line added.
- The default behaviour of the ENTER key has changed. When the ENTER key is hit while in the prefix area, if there are any pending prefix commands, they are executed, otherwise the cursor is moved to the first line of the next line as it used to do.
- [SET] TABSIN ON in the profile file will not increment the ALT count, nor include the changed lines in the recovery list.
- The SOS ENDCHAR command was incorrect in that it would move the cursor to the start of a line if the cursor was past the end of the line as well as moving the cursor to the end of the line if it wasn't past the end of the line. The correct behaviour of SOS ENDCHAR is to move to the end of the focus line no matter where the cursor currently is. For those of you (like me) who are used to the old behaviour of SOS ENDCHAR, a new SOS command; STARTENDCHAR has been added to THE.
- [SET] BACKUP now has TEMP and KEEP options.
- The FILE command behaviour has been changed. Previous versions would not write out a new version of the file if the alteration count was zero and no filename was specified. FILE now behaves the same as XEDIT and KEDIT.
- Added ability to specify individual tab columns in SET TABS command.

New features:

- Changed the method of displaying the screen. This was done to cater for shadow, tab,scale and reserved lines and to increase performance.
- Added "−r" command line switch to operate THE in readonly mode. This is a real readonly mode; it is not possible to change the contents of the file.
- Repeating targets with boolean operators are now supported.
- The targets BLANK and ~BLANK are now supported.
- A command can now follow a target on the command line without requiring a linend character separating the target and command.

```
eg. :3 del
is a valid command. You can even enter
3 3 3 3
on the command line to move the current line 12 lines.
```

- Colour settings are now stored with the file, so different files can have different colours.
- Wordwrap behaves like KEDIT wordwrapping. When the right−most character of the line exceeds the right margin, the last word wraps to the next line.
- No limit (apart from available memory) on the number of prefix commands that may be entered. Used to be a limit of 20.
- Key names no longer need to be expressed in exact case. C−A and c−a are equivalent. Also, where appropriate, keys have the same name across platforms.
- THE now can be compiled with AIX Extended Curses. This enables colour support when run in an aixterm window.

Version 1.5 01−Dec−93

Bug fixes:

- It is now possible to issue the commands; EDIT,THE and XEDIT from within a macro or profile file.
- [SET] CASE settings now inherited by subsequent files editted.
- [SET] CASE now does not reset settings to default if they are not specified; it leaves the values as they were last set.
- default value for ZONE end is now the maximum line width not 32766 (maximum possible line length)
- Bug in CHANGE command. If specifying a trailing space to change a string at the end of a line to null, one too many characters were removed. eg. c/_N // * 1 a line ending in "abc_N" ended up ending in "ab" When changing text at the end of a line and specify more than one space at the end of the target, the target is never found. eg. c/_N // * 1 a line ending in abc_N will not be found Handling of spaces after the real end of the line handled correctly now.
- Fixed a bug on Unix systems where a key that returned nul would execute 'add 1'.
- Changed the default handling of 'sos delback' and 'sos delchar' under UNIX. The default key sequences of these keys has been reversed. If you prefer to keep the old definitions, define OLD_DELCHAR_DELBACK on the compile line. This option will disappear in a future version so if you are really attached to the old behaviour, you had better let me know and have a good reason :−) Thanks to Andreas Schott for pointing this out!
- Added SRC line to makefile for SystemVR4 block.
- The default value for THE_HOME_DIR under Unix is /opt/THE, if SYSVR4 is defined when compiling or /usr/local/THE otherwise. The major change is the uppercasing of THE (THE's correct acronym)
- [SET] MACROPATH now sets the correct path value.
- Fixed bug with prefix command like 'aa'. The A command would be executed with a large number of lines added or you would get a core dump :−(
- After bringing a suspended THE session to the foreground, THE correctly refreshes.
- When shelling out, the screen attributes are returned to NORMAL, at least on some platforms;
- Fixed the occasional strange behaviour of the command line having some of the contents of the prefix area. (It was VERY obscure!)
- SOS DELWORD would increment ALT and not allow SOS UNDO to work.
- C−ENTER now valid for DOS and OS/2 versions.
- Ignore any command issued from a macro file if no more files are left in the ring.
- Changed the method for splitting a path and filename under DOS and OS/2. Hopefully fewer problems with ending up in the wrong directory. This has also worked around the bug in BCOS2.
- Reexecute command,= should now reexecute the last statement correctly.
- Printing under Unix more than once now works.
- Fixed a few bugs with PUT and GET.

New commands:

- Added TERMINAL, LASTMSG, MONITOR, POINT [*], PREFIX [SYNONYM name|*] PENDING, GETENV and BLOCK extract options.
- Added REXX macro for summing the contents of a marked BOX block.
- Added REXX macro for matching paired words like #ifdef/#endif.
- Added SORT command.
- Added [SET] LINEND command to support multiple commands on a line.
- Added [SET] ETMODE to indicate if extended characters should be displayed. This is designed to handle foreign languages that use the ASCII characters > 127.
- Added [SET] NONDisp to specify what character is displayed if extended characters are not to be displayed.
- Added [SET] PENDing to assist with writing block prefix macros.

- Added [SET] IMPOS/IMPcmscp to allow operating system commands to be executed from the command line without the need to prefix the command with OS or !.
- Added COMMAND command (Just for Ian Collier ;−))

Changed commands:

- Added extra argument to [SET] CASE to determine case significance in SORT command.
- The order of EXTRACT /VERSION/ and QUERY VERSION options has changed.
- BOTTOM command makes the last line of the file the current line instead of the *** Bottom of file *** line.
- INPUT [text] command results in the line becoming the current line instead of the previous line, when issued from the command line.
- Overhaul of commands that add a new line to fix a few bugs.
- STATUS now displays in 3 columns; I was starting to run out of space with 2 columns. The item name is also highlighted.
- When compiled with REXX support, [SET] PREFIX now supports prefix synonyms.
- PUT and PUTD now work with BLOCK as the target.
- [SET] TABKey changed to be simpler.
- [SET] ARBchar, TABSIn, TABSOut changed to allow their second parameters to be specified with the OFF option.
- STATus command now has optional filename parameter.
- [SET] TABS now has 'INCR' as first operand when EXTRACTing or Querying the value.
- MODIFY/QUERY of CMDLINE does not contain 2nd operand. This value is only returned as a result of EXTRACT.
- There is no need to supply a final '/' with the EXTract command.

New features:

- REXX support for DOS is now available. This version, supplied as a separate archive, contains the THE executable linked with a DOS port of Regina. This version contains a DOS extender, so only runs on 386s or above. This version was compiled with DJGPP, a DOS port of GNU gcc. Included is go32.exe, which is the DOS extender.
- By default all THE macros are expected to have a file name ending in ".the". The value for the macro file extension can be changed with the [SET] MACROEXT command. This default extension is only applicable to macros searched for in the THE_MACRO_PATH.
- Added defines.h to specify limits which can be safely changed in the source code.
- Added ability to execute multiple commands from command line and to assign multiple commands to a key.
- Now have the option to display extended ASCII codes on Unix platforms.
- Prefix macros can now be executed. Thanks to Dave Rittenhouse (david@ecst.csuchico.edu) for adding this feature.
- Prefix synonyms are also definable for REXX macros and standard prefix commands.
- THE now compiles and runs with BCOS2.
- Added −n command line switch to enable THE to be run without any profile file.

Warnings:

- There is an inconsistancy with Regina 0.05h in that the use of implied extract with some functions will corrupt values in THE. Implied extract functions that have been observed to corrupt, are fname.1() and fpath.1().

Version 1.4 01−Sep−93 (Not officially released)

Bug fixes:

- Changed the method for determining the filename of a backup file under OS/2. Originally, if drive type not = FAT, .bak was appended to full file name. Now only if drive type = HPFS is .bak appended.
- Alteration count was being set to zero when saving a file and the file could not be saved.
- Fixed a few minor bugs in DUPLICATE command.

New commands:

- Added support for semi−colon specifying an absolute line target

New features:

- Added support for ncurses under linux. ncurses still has a bug (I believe) that causes some characters to not be cleared.
- emx 0.8g port now has REXX support
- DUPLICATE 1 BLOCK assigned to Alt−D (DOS/OS2) and ControlD (UNIX)

Version 1.3 15−Aug−93

Bug fixes:

- Fixed bug with line block delete and cursor positioning.
- number of files being editted no longer gets out of sync when an attempt to edit a new file is aborted (usually because line is too long)
- Fixed bug in uppercase/lowercase when target was "BLOCK".
- sos tabwordb, sos tabwordf, and sos delword now correctly go the start/end of the word if that word exceeds the width of the screen.

New commands:

- SOS CURSORAdj
- SOS DELWord
- [SET] MARgins
- [SET] WORDWrap
- SPlit
- Join

New features:

- Changed the highlighting of "non−printable characters" under Unix.
- emx 0.8g compiler support now working (excluding REXX)
- wordwrap has been added

Changes:

- The names of environment variables and the location of default help and profile files has changed. See the file appendix.1 or Appendix 1 of the.man for details.

Version 1.2 27−Jun−93 (Not officially released)

Bug fixes:

- Previous command line contents were remaining when 'sos undo' was executed on command line.
- THE core dumped when a 'DELETE' command was present in a profile file.
- THE now displays line numbers > 32k correctly.
- Display of long filename in window wider than 80 chars now correct.
- [SET] CURLine now works when called from a profile file

New features:

- THE now handles either forward (/) or back (\) slashes in path names and converts them automatically. So you can specify d:/path under DOS and OS/2 or \usr\bin under Unix. All paths are displayed on the idline with the default OS path separator.
- added new external function, valid_target() to simplify REXX macros
- added new single character indicator on right end status line to indicate if REXX support is present. Meanings of indicators:

```
First character: (colour support)
  C − curses library supports colour and so does monitor
  c − curses library supports colour but monitor doesn't
  M − curses library does not support colour
Second character: (REXX support)
  R − THE compiled with REXX support
    − (blank) THE compiled without REXX support
```

Version 1.1 17−May−93

Bug fixes:

- A file with no end of line character on the last line, the last line would not be included in the file.
- Characters that are in blocks retain their highlighting now.
- Fixed schange under DOS and OS/2. Cursor is now positioned correctly.
- Writing out a file will now produce an error if the disk is full and not write part of the file.
- GET and PUT now recognise ~ in Unix version.
- PREFIX ON now no longer requires the LEFT|RIGHT option. Defaults to LEFT.
- TABPRE (to tab between main and prefix areas) now does nothing if PREFIX is OFF.
- 'bleeding' through of one file to another on BSDish systems now fixed.

New commands:

- BOX BLOCK COPY/MOVE/DELETE/FILL/OVERLAY
- LINE BLOCK COPY/MOVE/DELETE
- FILLBox
- sos_makecurr
- [SET] DIRInclude
- [D]OSNowait/[D]OSQuiet
- = (re−execute command)
- [SET] CMDArrows
- [SET] CMDline
- [SET] NEWlines
- [SET] MSGMode

- [SET] MACROPath
- [SET] IMPMACro
- [SET] NUMber
- [SET] HEX
- [SET] CLEARScreen
- [SET] Point
- [SET] REXXOUTput
- [SET] CLOCK
- [SET] HEXDISPlay
- SOS DOPREfix
- the target ALL has been implemented. This is NOT the ALL command.
- MACRO − execute commands from a file
- UPPercase, LOWercase
- SHift
- DUPlicate
- EXPand
- STATus
- Query
- EMSG
- SUSPend
- REDRAW
- MODIFY
- TEXT
- prefix commands: a,i,c,m,d,",,/,cc,mm,dd,"",<>,.xxxxx

Changed commands:

- insertmode now requires a parameter: ON|OFF|TOGGLE
- spltjoin now correctly aligns the new line under the focus line
- sos_* commands have been changed to separate sos commands
- REFRESH command now consistant with XEDIT and KEDIT. Use REDRAW for old REFRESH functionality.

New features:

- REXX support under OS/2 and Unix(with Regina 0.05)
- argument passing to REXX macros/profiles
- EXTRACT
- Capture REXX trace and Say output to a file in the ring.

Commands removed (temporarily)

- SCREEN

Version 1.0 16−Aug−92

Released

- First release to the unsuspecting public.
- Sent to SIMTEL and comp.binaries.os2.

Generated on: 2 Nov 2002

Return to Table of Contents

THE Quick Reference
Version 3.1

Command−line invocation

the [−h?nmrsbq] [−p profile] [−a prof_arg] [−l line_num] [−c col_num]

Where:

−h or −?	Display usage information and version.
−n	Do not execute a profile file.
−m	Force display into mono, ignoring colour support.
−r	Run THE in read−only mode.
−s	A debugging aid for Unix versions. Turns off signal trapping.
−b	run in batch mode.
−q	run quietly in batch mode
−k[fmt]	allow Soft Label Key display
−l line_num	specify line number to make current
−c column_num	specify column number to make current
−p profile	The name of the profile file to use. If not specified, the file ".\PROFILE.THE" (under OS/2, DOS and Win95/NT) or "$HOME/.therc" (under UNIX), will be used, unless overridden by the environment variable "THE_PROFILE_FILE".
−a prof_arg	Used to pass a string to the profile file as argument. Only works with Rexx macros.
−w width	Length of longest line THE can handle (default 512)
−u disp_len	display length in non−line mode
dir	Will display a directory of the specified dir.
file [...]	Specifications of file(s) to edit. If multiple files are specified, each will be put in a separate file in the ring. The last file specified will be displayed. If no files are specified, the current directory will be displayed. Issuing [SOS EDIT] on a file spec will edit that file. [SOS EDIT] on a subdir will display that directory.

Option flags are case−sensitive.

Minimum Abbreviations

In the following segments, the capitalised portion of a command is the minimum abbreviation for the command.

Commands

Add [n]
add blank line
Default: 1

ALERT /prompt/ [EDITfield [/val/]] [TITLE /title/] [OK|OKCANCEL|YESNO|YESNOCANCEL] [DEFBUTTON n]
display a user configurable dialog box with notification
ALL [rtarget]
select and display restricted set of lines
BAckward [n|*]
scroll backwards [n] screens
Default: 1
Bottom
move to the bottom of the file
CANcel
quit from all unaltered files in the ring
CAppend [text]
append text after column pointer
CCancel
qquit from all files in the ring
CDelete [column target]
delete text starting at column pointer
CFirst
move column pointer to beginning of zone
Change [/string1/string2/ [target] [n] [m]]
change one string to another
Default: 1 1 1
CInsert text
insert text starting at the column pointer
CLAst
move the column pointer to end of zone
CLocate column target
move the column pointer
CMATCH
find matching bracket character
CMSG [text]
display text on command line
COMMAND command [options]
execute a command without translation
COMPress [target]
reduce spaces to tabs
CONTROLChar
allow control characters to be entered
COPY target1 target2
COPY BLOCK [RESET]
copies text from one position to another
COVerlay text
overlay text starting at the column pointer
CReplace text
replace text starting at the column pointer
CURsor Column
CURsor Screen UP|DOWN|LEFT|RIGHT
CURsor Screen row [col]
CURsor [Escreen] UP|DOWN
CURsor [Escreen|Kedit] LEFT|RIGHT
CURsor [Escreen] row [col]
CURsor CUA UP|DOWN|LEFT|RIGHT
CURsor CMdline [n]

CURsor HOME [SAVE]
CURsor File line [col]
CURsor GOTO line col
CURsor Mouse
move cursor to specified position
DEFine key−name [REXX] [command [args] [[#command [args]...]]]
DEFine mouse−key−definition IN window [REXX] [command [args] [[#command [args]...]]]
assign one or many commands to a key or mouse event
DELete [target]
delete lines from a file
Default: 1
DIALOG /prompt/ [EDITfield [/val/]] [TITLE /title/] [OK|OKCANCEL|YESNO|YESNOCANCEL] [DEFBUTTON n]
display a user configurable dialog box
DIRectory [file specification]
list the specified directory as an editable file
DOS [command]
execute an operating system command
DOSNowait command
execute an operating system command − no prompt
DOSQuiet command
execute an operating system command quietly
Down [relative target]
move forward in the file a number of lines
Default: 1
DUPlicate [n [target|BLOCK]]
duplicate lines
Edit [file]
edit another file or switch to next file
EDITV GET|PUT|GETF|PUTF var1 [var2 ...]
EDITV SET|SETF var1 value1 [var2 value2 ...]
EDITV SETL|SETLF|SETFL var1 value1
EDITV LIST|LISTF [var1 ...]
set and retrieve persistent macro variables
EMSG [message]
display message
enter [CUA]
execute a command
EXPand [target]
expand tab characters to spaces
EXTract /item/[...]
obtain various internal information about THE
FFile [filename]
force a FILE of the current file to disk
Default: With no parameters, the current file is written.
FILE [filename]
write the current file to disk and remove from ring
Default: With no parameters, the current file is written.
FILLbox [c]
fill the marked block with a character
Find [string]
locate forwards the line which begins with the supplied string
FINDUp [string]
locate backwards the line which begins with the supplied string

FOrward [n]
scroll forwards [n] screens
Default: 1
FUp [string]
locate backwards the line which begins with the supplied string
GET [filename] [fromline] [numlines]
GET CLIP: [STREAM|BOX|LINE]
insert into file the contents of specified file
HELP
edit help file for THE
HIT key
simulate hitting of the named key
Input [string]
insert the command line contents into the file
Join [ALigned] [Column|CURSOR]
join a line with the line following
Kedit [file]
edit another file or switch to next file
LEft [n|HALF]
scroll the screen to the left
[Locate] target [command]
search for a target
LOWercase [target]
change uppercase characters to lowercase
LS [file specification]
list the specified directory as an editable file
MACRO [?] filename [arguments ...]
execute a macro command file
MARK Box [line1 col1 line2 col2]
MARK Line [line1 line2]
MARK Stream [line1 col1 line2 col2]
MARK Column [col1 col2]
MARK Word [line1 col1]
MARK CUA [LEFT|RIGHT|UP|DOWN|START|END|FOrward|BAckward|TOP|Bottom|MOUSE]
mark a portion of text
MODify set−command
display current SET command for alteration
MOVE target1 target2
MOVE BLOCK [RESET]
move a portion of text
MSG [message]
display message on error line
Next [relative target]
move forward in the file a number of lines
Default: 1
NEXTWindow
switch focus of editing session to another file
NFind [string]
locate forwards the line which does NOT begin with the supplied string
NFINDUp [string]
locate backwards the line which does NOT begin with the supplied string
NFUp [string]
locate backwards the line which does NOT begin with the supplied string
NOMSG command [arguments]

execute a command suppressing any messages
NOP
no operation command
OS [command]
execute an operating system command
OSNowait command
execute an operating system command − no prompt
OSQuiet command
execute an operating system command quietly
OSRedir filename command [arguments ...]
execute an operating system command and capture output
OVERLAYBox
overlay marked block on current cursor position
POPUP [MOUSE|TEXT|CENTER|CENTRE|BELOW|ABOVE] [INITIAL n] [ESCAPE keyname] /item1[/item2/...]
display popup menu
PRESERVE
save various editor settings
PREVWindow
switch focus of editing session to another file
PRint [target] [n]
PRint LINE [text]
PRint STRING [text]
PRint FORMfeed
PRint CLOSE
send text to default printer or print spooler
PUT [target] [filename]
write part of a file to another
PUTD [target] [filename]
write part of a file to another and delete
QQuit
exit from the current file without saving changes
Query item
display various option settings
QUIT
exit from the current file if no changes made
READV Cmdline [initial text]
READV KEY
read keystrokes and pass to macro
RECover [n|*]
recover changed or deleted lines
REDIT
re−edit the current file
REDRAW
redraw the current screen
REFRESH
refresh the contents of the current screen
REPEat [target]
repeat the last command
Replace [text]
replace the current line with supplied text
RESet ALL|Block|Prefix|THIGHlight
cancel the marked block or prefix commands or both
REStore

restore various editor settings
REXX rexx instructions
execute Rexx instructions
RGTLEFT [n]
scroll the screen to the left or right
RIght [n|HALF|FULL]
scroll the screen to the right
SAVE [filename]
save changes to current file
SCHange /string1/string2/ [target] [n] [m]
selectively change strings
Default: 1 1 1
SET set_command [set_command parameter(s) ...]
execute various set commands
SHift Left|Right [n] [target]
move text left or right
SHOWkey [ALL]
display current key value and command assignation
SORT target [[sort field 1] [...] [sort field 10]]
sort selected lines in a file
SOS sos_command [sos_command ...]
execute various sos commands
SPlit [ALigned] [Column|CURSOR]
split a line into two lines
spltjoin
split/join two lines
SSave [filename]
force SAVE to specified file
STATus [filename]
display current settings of various variables
SUSPend
suspend THE and return to operating system
TABFILE [+|−]
edit the file under the file tab or shift FILETABS view
tabpre
switch between FILEAREA and PREFIX area
TAG [More|Less] [rtarget]
displays lines matching target in different colour
TEXT text
simulate keyboard entry of characters
THE [filename]
edit another file or switch to next file
TOASCII [target]
convert the target from EBCDIC to ASCII
TOP
move to the top of the file
Up [relative target]
move backward in the file a number of lines
Default: 1
UPPercase [target]
change lowercase characters to uppercase
Xedit [file]
edit another file or switch to next file
?[+|?...]

retrieve − return the next/prior command on the command line
=
re−execute the last command issued on the command line
! [command]
execute an operating system command

execute and re−display command

SET commands

[SET] ALT [n] [m]
change alteration counts
Default: OFF
[SET] ARBchar ON|OFF [char1] [char2]
set arbitrary character(s) for targets
Default: Off $?
[SET] AUTOCOLOR mask parser [MAGIC]
specifies which parser to use for syntax highlighting
Default: See AUTOCOLOR
[SET] AUTOCOLOR mask parser [MAGIC]
specifies which parser to use for syntax highlighting
Default: See AUTOCOLOR
[SET] AUtosave n|OFF
set autosave period
Default: OFF
[SET] AUTOSCroll n|OFF|Half
set rate of automatic horizontal scrolling
Default: HALF
[SET] BACKup OFF|TEMP|KEEP|ON|INPLACE
indicate if a backup copy of the file is to be kept
Default: KEEP
[SET] BEEP ON|OFF
turn on or off the audible alarm when displaying errors
Default: OFF
[SET] BOUNDMARK OFF|Zone|TRunc|MARgins|TABs|Verify
set bounds marker display
Default: Zone
[SET] CASE Mixed|Lower|Upper [Respect|Ignore] [Respect|Ignore] [Respect|Ignore]
set case sensitivity parameters
Default: Mixed Ignore Respect Respect
[SET] CLEARErrorkey *|keyname
specify which key clears the message line
Default: *
[SET] CLEARScreen ON|OFF
indicate if the screen is to be cleared on exit
Default: OFF
[SET] CLOCK ON|OFF
turn on or off display of time on status line
Default: ON
[SET] CMDArrows Retrieve|Tab
sets the behaviour of the up and down arrow keys
Default: RETRIEVE
[SET] CMDline ON|OFF|Top|Bottom
sets the position of the command line.

Default: BOTTOM
[SET] COLOR area [modifier[...]] [foreground] [ON] [background]
[SET] COLOR area [modifier[...]] ON|OFF
set colours for display
Default: Depends on compatibility mode setting and monitor type.
[SET] COLOUR area [modifier[...]] [foreground] [on background]
[SET] COLOUR area [modifier[...]] ON|OFF
set colours for display
Default: Depends on compatibility mode setting and monitor type.
[SET] COLORING ON|OFF [AUTO|parser]
enable or disable syntax highlighting
Default: ON AUTO
[SET] COLOURING ON|OFF [AUTO|parser]
enable or disable syntax highlighting
Default: ON AUTO
[SET] COMPat The|Xedit|Kedit|KEDITW|Ispf|= [The|Xedit|Kedit|KEDITW|Ispf|=]
[The|Xedit|Kedit|KEDITW|Ispf|=]
set compatibility mode
Default: THE THE THE
[SET] CTLchar OFF
[SET] CTLchar char Escape | OFF
[SET] CTLchar char Protect|Noprotect [modifier[...]] [fore [ON back]]
define control character attributes
Default: OFF
[SET] CURLine M[+n|−n] | [+|−]n
set position of current line on screen
Default: M
[SET] CURSORSTay ON|OFF
set on or off the behaviour of the cursor on a scroll
Default: ON
[SET] DEFSORT OFF|DIRectory|Size|Date|Time|Name [Ascending|Descending]
specify the order in which files appear in DIR.DIR
Default: NAME ASCENDING
[SET] DIRInclude *
[SET] DIRInclude [Normal] [Readonly] [System] [Hidden] [Directory]
set the file mask for directory command
Default: *
[SET] DISPlay n [m|*]
specify which level of lines to display
Default: 0 0
[SET] ECOLOR char [modifier[...]] [foreground] [on background]
[SET] ECOLOR char [modifier[...]] ON|OFF
set colors for syntax highlighting
Default: See ECOLOR
[SET] ECOLOUR char [modifier[...]] [foreground] [on background]
[SET] ECOLOUR char [modifier[...]] ON|OFF
set colours for syntax highlighting
Default: See ECOLOR
[SET] EOLout CRLF|LF|CR|NONE
set end of line terminating character(s)
Default: LF − UNIX
Default: CRLF − DOS/OS2/WIN32
Default: NONE − if THE started with −u option
[SET] EQUIVChar char

set the equivalence character
Default: =
[SET] ETMODE ON|OFF [character list]
indicate if extended display mode is possible
Default: ON − DOS/OS2/WIN32
Default: ON 32−255 − X11
Default: OFF − UNIX/AMIGA/QNX
[SET] FExt ext
[SET] FType ext
change the extension of the existing file
[SET] FILEName filename
change the filename of the file being edited
[SET] FILETABS ON|OFF
determine if and where where file tabs are positioned
[SET] FMode d[:]
change the drive letter of the existing file
[SET] FName filename
change the filename of the file being edited
[SET] FPath path
change the path of the existing file
[SET] FType ext
change the extension of the existing file
[SET] FULLFName ON|OFF
specify if complete filename to be displayed
Default: ON
[SET] HEADer section ON|OFF
turn on or off syntax highlighting headers
Default: * ON
[SET] HEX ON|OFF
set how hexadecimal strings are treated in string operands
Default: OFF
[SET] HEXDISPlay ON|OFF
turn on or off display of character under cursor
Default: ON
[SET] HEXShow ON|OFF [M[+n|−n]|[+|−]n]
turn on or off hex display of current line
Default: OFF 7
[SET] HIGHlight OFF|TAGged|ALTered|SELect n [m]
specify which lines (if any) are to be highlighted
Default: OFF
[SET] IDline ON|OFF
specify if IDLINE is displayed
Default: ON
[SET] IMPcmscp ON|OFF
set implied operating system command processing
Default: ON
[SET] IMPMACro ON|OFF
set implied macro command processing
Default: ON
[SET] IMPOS ON|OFF
set implied operating system command processing
Default: ON
[SET] INPUTMode OFF|FUll|LIne
set input mode behaviour

Default: LINE
[SET] INSERTMode ON|OFF|TOGGLE
put editor into or out of insert mode
Default: OFF
[SET] INTerface CLASSIC|CUA
set overall behaviour of THE
Default: CLASSIC
[SET] LINEFLAG CHAnge|NOCHange NEW|NONEW TAG|NOTAG [target]
set the line characteristics of lines
Default: NOCHANGE NONEW NOTAG
[SET] LINENd ON|OFF [character]
allow/disallow multiple commands on command line
Default: OFF #
[SET] MACRO ON|OFF
indicate if macros executed before commands
Default: OFF
[SET] MACROExt [ext]
set default macro extension value
Default: the
[SET] MACROPath PATH|path[s]
set default path for macro commands
Default: Path specified by env variable THE_MACRO_PATH
[SET] MARgins left right [[+|−]indent]
set left and right margins for wordwrap
Default: 1 72 +0
[SET] MOUSE ON|OFF
turn mouse support on or off
Default: ON − if mouse supported, OFF − otherwise
[SET] MSGLine ON M[+n|−n]|[+|−]n [lines] [Overlay]
[SET] MSGLine CLEAR
set position and size of message line
Default: ON 2 5 Overlay
[SET] MSGMode ON|OFF
set display of messages on or off
Default: ON
[SET] NEWLines Aligned|Left
set position of cursor after adding blank line
Default: Aligned
[SET] NONDisp character
specify character to display for non−displaying characters
Default: #
[SET] NUMber ON|OFF
turn prefix numbers on or off
Default: ON
[SET] PAGEWRAP ON|OFF
determine if page scrolling wraps at bottom/top of file
Default: ON
[SET] PARSER parser file
associates a language definition file with a parser
[SET] PENDing ON string
[SET] PENDing OFF
[SET] PENDing BLOCK string
set status of pending prefix commands
[SET] Point .name [OFF]

assign a name to the current line
[SET] POSition ON|OFF
determine if LINE/COL is displayed on idline
Default: ON
[SET] PREfix ON [Left|Right] [n [m]]
[SET] PREfix Nulls [Left|Right] [n [m]]
[SET] PREfix OFF
[SET] PREfix Synonym newname oldname
set prefix area attributes
Default: ON Left 6 0
[SET] PRINTER spooler|[OPTION options]
define printer spooler name
Default: LPT1 − DOS/OS2, lpr − Unix, default − Win32
[SET] PSCReen height [width] [RESET|PRESET]
set physical size of screen
Default: System Dependent
[SET] READONLY ON|OFF|FORCE
allow/disallow changes to a file if it is readonly
Default: OFF
[SET] REGEXP syntax
specify the regular expression syntax to use
Default: EMACS
[SET] REPROFile ON|OFF
indicate if profile file to be executed for all files
Default: OFF
[SET] RESERved *|+|−n [colour] [text|OFF]
display a reserved line
[SET] REXXOUTput File|Display n
indicate where Rexx output is to go
Default: Display 1000
[SET] SCALe ON|OFF [M[+n|−n]|[+|−]n]
set position and status of scale line on screen
Default: OFF M+1
[SET] SCOPE All|Display
sets which lines are to be excluded from commands
Default: Display
[SET] SCReen n [Horizontal|Vertical]
[SET] SCReen Size l1|* [l2|*]
specify number of screens displayed
Default: 1
[SET] SELect [+|−]n [target]
sets the selection level for the specified lines
Default: 0
[SET] SHADOW ON|OFF
determines if shadow lines are displayed or not
Default: ON
[SET] SLK n|OFF [text]
set Soft Label Key definitions
Default: OFF
[SET] SPAN ON|OFF [Blank|Noblank [n|*]]
specify if a string target can span multiple lines (unavailable)
Default: OFF Blank 2
[SET] SPILL ON|OFF|WORD
specify if a string target can span multiple lines (unavailable)

Default: OFF
[SET] STATOPT ON option column [length [prompt]]
[SET] STATOPT OFF option|*
set display options on statusline
Default: ON NBFILE.1 13 0 Files=
Default: ON WIDTH.1 23 0 Width=
[SET] STATUSLine Top|Bottom|Off|GUI
set position of status line
Default: Bottom
[SET] STAY ON|OFF
set condition of cursor position after CHANGE/LOCATE commands
Default: ON
[SET] SYNonym ON|OFF
[SET] SYNonym [LINEND char] newname [n] definition
define synonyms for commands (unavailable)
Default: OFF
[SET] TABKey Tab|Character Tab|Character
set characteristics of the SOS TABF command
Default: Tab Character
[SET] TABLine ON|OFF [M[+n|−n]|[+|−]n]
set position and status of tab line on screen
Default: OFF −3
[SET] TABS n1 [n2 ... n32]
[SET] TABS INCR n
[SET] TABS OFF
set tab columns or tab length
Default: INCR 8
[SET] TABSIn ON|OFF [n]
set tab processing on file input
Default: OFF 8
[SET] TABSOut ON|OFF [n]
set tab processing on file output
Default: OFF 8
[SET] TARGETSAVE ALL|NONE| STRING REGEXP ABSOLUTE RELATIVE POINT BLANK
set type(s) of targets to save for subsequent LOCATEs
Default: ALL
[SET] THIGHlight ON|OFF
specify if text highlighting is supported
Default: ON − THE/KEDIT/KEDITW OFF − XEDIT/ISPF
[SET] TIMECHECK ON|OFF
specify if time stamp checking done
Default: ON
[SET] TOFEOF ON|OFF
specify if TOF and BOF lines are displayed
Default: ON
[SET] TRAILING ON|OFF|SINGLE|EMPTY
specify the truncation column
Default: ON
[SET] TRunc n|*
specify the truncation column
Default: *
[SET] TYPEAhead ON|OFF
set behaviour of screen redraw
Default: OFF

[SET] UNDOING ON|OFF
turn on or off undo facility for the current file
Default: ON
[SET] UNTAA ON|OFF
specifies if "Unsigned Numerical Targets Are Absolute"
Default: OFF
[SET] Verify first [last]
set column display limits
Default: 1 *
[SET] WIDTH n
set width of maximum line that THE can edit
Default: 512
[SET] WORD NONBlank|ALPHAnum
controls what THE considers a word to be
Default: NONBlank
[SET] WORDWrap ON|OFF
set wordwrap feature on or off
Default: OFF
[SET] WRap ON|OFF
enable/disable string locates around the end of the file
Default: OFF
[SET] XTERMinal program
set X terminal to execute under X
Default: System dependent but usually one of:
[SET] Zone first [last]
set column limits for editing
Default: 1 *

SOS commands

SOS ADDline
add blank line after focus line
SOS BLOCKEnd
move cursor to end of marked block
SOS BLOCKStart
move cursor to start of marked block
SOS BOTTOMEdge
move cursor to bottom edge of FILEAREA
SOS CUADELBAck
delete the character to the left of the cursor
SOS CUADELChar
delete character under cursor
SOS CURRent
move cursor to current line
SOS CURSORAdj
move first non−blank character to cursor
SOS CURSORSHIFT
move text to right of cursor to cursor
SOS DELBAck
delete the character to the left of the cursor
SOS DELChar
delete character under cursor
SOS DELEnd
delete to end of line

SOS DELLine
delete focus line
SOS DELWord
delete word at or right of cursor
SOS DOPREfix
execute any pending prefix commands
SOS EDIT
edit a file from directory list
SOS ENDChar
move cursor to end of focus line
SOS EXecute
move cursor to command line and execute command
SOS FIRSTCHar
move cursor to first non−blank of field
SOS FIRSTCOl
move cursor to first column of field
SOS INSTAB
shift text to next tab column
SOS LASTCOl
move cursor to last column of field
SOS LEFTEdge
move cursor to left edge of window
SOS LINEAdd
add blank line after focus line
SOS LINEDel
delete focus line
SOS MAKECURR
make focus line the current line
SOS MARGINL
move cursor to the left margin column
SOS MARGINR
move cursor to the right margin column
SOS PARINDent
move cursor to the paragraph indent column
SOS PASTECMDline
copy contents of marked block to command line
SOS PREfix
move cursor to leftmost edge of prefix area
SOS QCmnd
move cursor to command line and clear
SOS RIGHTEdge
move cursor to right edge of window
SOS SETTAB
set a tab column at the cursor position
SOS STARTENDChar
move cursor to end/start of focus line
SOS TABB
move cursor to previous tab stop
SOS TABf
move cursor to next tab stop
SOS TABFIELDB
move cursor to previous enterable field
SOS TABFIELDf
move cursor to next enterable field

SOS TABWORDB
move cursor to beginning of previous word
SOS TABWORDf
move cursor to start of next word
SOS TOPEdge
move cursor to top edge of filearea
SOS UNDO
undo changes to the current line

Prefix commands

a	target − after line	add blank line
b	target − before line	N/A
i	insert blank line	insert blank line
/	N/A	make line current line
c	copy line	copy line
m	move line	move line
d	delete line	delete line
x	exclude line	exclude line
"	N/A	duplicate line
r	repeat line	N/A
o	overlay line	overlay line
s	show excluded lines	show excluded lines
p	N/A	target − prior to line
f	N/A	target − following line
>	shift line right	shift line right
<	shift line left	shift line left
)	shift line right zoned	shift line right zoned
(	shift line left zoned	shift line left zoned
lc	lowercase line	lowercase line
uc	uppercase line	uppercase line
cc	copy line block	copy line block
mm	move line block	move line block
dd	delete line block	delete line block
xx	exclude line block	exclude line block
""	N/A	duplicate line block
rr	repeat line block	N/A
oo	overlay line block	overlay line block
>>	shift line block right	shift line block right
<<	shift line block left	shift line block left
))	shift line block right zoned	shift line block right zoned
((	shift line block left zoned	shift line block left zoned
lcc	lowercase line block	lowercase line block
ucc	uppercase line block	uppercase line block
.xxxxx	give name to line	give name to line
TABL	N/A	display tab line
SCALE	N/A	display scale line
TABS	display tab line	N/A
COLS	display scale line	N/A

Line Targets

Absolute target:	An absolute line number. eg. :12 = line number 12
All target:	All lines in file the file. eg. ALL
Blank target:	The first line containing only blank characters. eg. BLANK
Block target:	All lines in the marked block. eg. BLOCK
Named line target:	A line with a name, set with [SET] POINT eg. .fred
Relative target:	A target relative to the current line. eg. 12 = 12 lines below the current line −4 = 4 lines above the current line * = all lines below the current line −* = all lines above the current line
String target:	A sequence of characters between delimiters. eg. /string/ − the first line below the current line containing "string" −/string/ − the first line above the current line containing "string"

Targets may be separated by the boolean operators "or '|'.
eg. /string1/ | /string2/
Targets may also be prefixed with '~' to signify boolean NOT.
eg. ~/string/

Default Key Assignments

?	F6	F6	F6	F6
backward 1	PGUP	PGUP	PrevScreen	PGUP
cmatch	F7	F7	F7	F7
controlchar	F11	F11	F11	F11
copy block	A−K	C−K	C−K	C−K
copy block reset	A−C	C−C	C−C	C−C
cursor down	CURD	CURD	DOWN	CURD
cursor home save	HOME	HOME	F16 (Do)	HOME
cursor left	CURL	CURL	LEFT	CURL
cursor right	CURR	CURR	RIGHT	CURR
cursor up	CURU	CURU	UP	CURU
delete block	A−G	C−G	C−G	C−G
duplicate 1 block	A−D	C−D	C−D	C−D
enter	ENTER	ENTER	C−M	NUMENTER,C−M
file	F3	F3	PF3	F3
fillbox	A−F	C−F	C−F	C−F
forward *	C−PGDN	C−PGDN		
forward 1	PGDN	PGDN	NextScreen	PGDN
help	F1	F1	PF1	F1
insertmode toggle	INS	INS	InsertHere	INS
join aligned	A−J	C−J	C−J	C−J
locate	S−F1	S−F1	F17	
locate .a	C−F12	C−F12		
mark box	A−B	C−B	C−B	C−B
mark line	A−L	C−L	C−L	C−L
move block reset	A−M	C−V	C−V	C−V

nextwindow	F2	F2	PF2	F2
overlaybox	A−O	C−O	C−O	C−O
point .a	C−F11	C−F11		
qquit	S−F3	S−F3	F13	
recover	F8	F8	F8	F8
redraw	C−R	C−R	C−R	C−R
reset block	A−U	C−U	C−U	C−U
sos addline 1	F4	F4	PF4	F4
sos cursoradj	C−A	C−A	C−A	C−A
sos delback	BKSP	BKSP	BackSpace	BACKSPACE
sos delchar	DEL	DEL	Remove,C−H	DEL
sos delend	C−END	C−END	C−E	C−E
sos delline 1	F9	F9	F9	F9
sos delword	A−W	C−W	C−W	C−W
sos doprefix	NUMENTER	NUMENTER	C−P	C−P
sos edit	A−X	C−X	C−X	C−X
sos makecurr	F5	F5	F20	F5
sos startendchar	END	END	Select	
sos tabb	S−TAB	S−TAB		
sos tabf	TAB	TAB	C−I	TAB
sos tabwordb	C−CURL	C−CURL	F18	
sos tabwordf	C−CURR	C−CURR	F19	
sos undo	ESC	ESC	C−Q	C−Q
split aligned	A−S	C−S	C−S	C−S
spltjoin	F10	F10	F10	F10
suspend	C−Z	C−Z	C−Z	C−Z
tabpre	F12,PLUS,NUMPLUS	F12,PLUS	Find,F12	F12
top	C−PGUP	C−PGUP		

Generated on: 2 Nov 2002

Return to Table of Contents

GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "fdl_html_free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "fdl_html_copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "fdl_html_Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "fdl_html_you".

A "fdl_html_Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "fdl_html_Secondary Section" is a named appendix or a front−matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "fdl_html_Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "fdl_html_Cover Texts" are certain short passages of text that are listed, as Front−Cover Texts or Back−Cover Texts, in the notice that says that the Document is released under this License.

A "fdl_html_Transparent" copy of the Document means a machine−readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "fdl_html_Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard−conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine−generated HTML produced by some word processors for output purposes only.

The "fdl_html_Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "fdl_html_Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front−Cover Texts on the front cover, and Back−Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine−readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly−accessible computer−network location containing a complete Transparent copy of the Document, free of added material, which the general network−using public has access to download anonymously at no charge using public−standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "fdl_html_History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "fdl_html_History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "fdl_html_History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "fdl_html_Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "fdl_html_Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "fdl_html_Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front−matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "fdl_html_Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties−−for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front−Cover Text, and a passage of up to 25 words as a Back−Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front−Cover Text and one of Back−Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "fdl_html_History" in the various original documents, forming one section entitled "fdl_html_History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "fdl_html_Dedications". You must delete all sections entitled "fdl_html_Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "fdl_html_aggregate", and this License does not apply to the other self−contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "fdl_html_or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
      Copyright (c)  YEAR  YOUR NAME.
      Permission is granted to copy, distribute and/or modify this document
      under the terms of the GNU Free Documentation License, Version 1.1
      or any later version published by the Free Software Foundation;
      with the Invariant Sections being LIST THEIR TITLES, with the
      Front−Cover Texts being LIST, and with the Back−Cover Texts being LIST.
      A copy of the license is included in the section entitled "fdl_html_GNU
      Free Documentation License"fdl_html_.
```

If you have no Invariant Sections, write "fdl_html_with no Invariant Sections" instead of saying which ones are invariant. If you have no Front−Cover Texts, write "fdl_html_no Front−Cover Texts" instead of "fdl_html_Front−Cover Texts being LIST"; likewise for Back−Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.