

File created: 28-Mar-2024 00:09:16 {WMEDLEY}<library>TEXEC.;6

edit by: rmk

changes to: (FNS TEXEC.DEFAULT.MENUFN TEXEC.SHRINK.ICONCREATE)

previous date: 18-Jun-2023 09:48:54 {WMEDLEY}<library>TEXEC.;5

Read Table: XCL

Package: INTERLISP

Format: XCCS

(RPAQQ TEXECOMS

```
((COMS ; To support development and compilation
(DECLARE\ : DONTCOPY EVAL@COMPILE (FILES TEDIT-EXPORTS.ALL (LOADCOMP)
ATERM)))

(COMS ;; THE FILLBUFFER REPLACEMENT CODE
(FNS TEXEC.BACKSKREAD TEXEC.OPENTEXTSTREAM TEXEC.DEFAULT.MENUFN TEXEC.DO?CMD TEXEC.CREATEMENU
TEXEC.GET TEXEC.INCLUDE TEXEC.FIND.FORWARD TEXEC.FIND.BACKWARD TEXEC.MENU.WHENHELDFN
TEXEC.SHRINK.ICONCREATE TEXEC.FILLBUFFER TEXEC.FILLBUFFER.TCLASS TEXEC.CHSELPENDING
TEXEC.FILLBUFFER.CHARDELETE TEXEC.FILLBUFFER.WORDDELETE TEXEC.FILLBUFFER.LINEDELETE
TEXEC.PARENCOUNT TEXEC.PARENMATCH TEXEC.FLASHCARET TEXEC.TEXTSTREAM.TO.LINEBUF TEXEC.FIX
TEXEC.NTHBUFCHARBACK TEXEC.NTHBACKCHNUM TEXEC.EOTP TEXEC.GETKEY TEXEC.INSERTCHAR TEXEC.DELETE
TEXEC.\CHDEL1 TEXEC.?EQUAL TEDIT.SCROLL? TEXEC.DISPLAYTEXT \TEXEC.TEXTBOUT
\TEXEC.TEXTBOUT1 \TEXEC.TEXTBOUT2 \TEXEC.TEXTBOUT3 \TEXEC.TEXTBOUT4 \TEXEC.SELFN
TEXEC.PRINTARGS TEXEC.PROCENTRYFN TEXEC.PROCEXITFN))

(COMS ;; Code to support a TEXEC lisp 'listener'
(FNS TEXEC TTEXEC)
(APPENDVARS (|BackgroundMenuCommands| (TEXEC '(TEXEC)
"Starts TEXEC in a new window."))))
(HORRIBLEVARS TEXEC.ICON TEXEC.ICON.MASK TEXEC.TITLED.ICON.TEMPLATE)
(VARS (TEXEC.ICON.TITLE.REGION (CREATE REGION BOTTOM _ 55 LEFT _ 6 WIDTH _ 77 HEIGHT _ 16))
(TEXEC.ICON.FONT (FONTCREATE 'HELVETICA 8 'BOLD))
(TEXEC.DEFAULT.MENU (TEXEC.CREATEMENU ' ((|Put| '|Put| NIL (SUBITEMS |Plain-Text| |Old-Format|))
(|Include| '|Include|)
("Forward Find" '|ForwardFind|)
("Backward Find" '|BackwardFind|)
(|Limit| '|Limit|))))
(|BackgroundMenu| NIL))
(INITVARS (TEXEC.BUFFERLIMIT 10000))))
```

;; To support development and compilation

```
(DECLARE\ : DONTCOPY EVAL@COMPILE
(FILESLOAD TEDIT-EXPORTS.ALL (LOADCOMP)
ATERM)
)
```

;; THE FILLBUFFER REPLACEMENT CODE

(DEFINEQ

(TEXEC.BACKSKREAD

```
(LAMBDA (BUFFER BUF NOTIFQUOTED) ; Edited 1-Dec-87 13:20 by jds
;; Returns buffer position of start of list containing cursor position BUF, or start of buffer. If NOTIFQUOTED is true, then returns NIL if the
;; paren/bracket at BUF is quoted with the escape char or is inside a string. Strategy: start at beginning of buffer and TTSKREAD forward (much
;; easier); if read ends at BUF, we win; if ends before BUF, then resume reading there (we skipped an internal list); otherwise if read did not end,
;; BUF must be inside a list, so scan ahead for start of an inner list, and repeat
(PROG ((B BUFFER)
(INNERMOSTLIST BUFFER)
(ESCAPED BRACKETFLG X)
LP (COND
((EQ B BUF) ; No list in buffer at all
(RETURN (AND (OR (NOT NOTIFQUOTED)
(NOT ESCAPED))
INNERMOSTLIST))))
(SELECTC (\SYNCODE \RDTBLSA (FIRSTCHAR B))
((LIST LEFTPAREN.RC LEFTBRACKET.RC) ; open paren or bracket. Try scanning this new internal list
(COND
(ESCAPED ; Inside a multiple escape
)
((EQ (SETQ X (TTSKREAD (CDR B)
BUF))
BUF)
(RETURN (OR BRACKETFLG B)))
(X ; Skip over internal list just scanned
(SETQ B X))
(T
;; The TTSKREAD failed, so BUF must be at least this deeply nested. Save pointer here in case we abort inside a
;; string or such
(SETQ INNERMOSTLIST B)
```

```

(COND
  ((AND (EQ (CAR B)
            (CHARCODE []))
        (EQ (CAR BUF)
            (CHARCODE []))))
  ;; Brackets may match; save position of this open bracket. Otherwise we'll return the innermost list, rather than
  ;; the start of the bracket expression
  (SETQ BRACKETFLG B))))))
(ESCAPE.RC
  (COND
    ((EQ (CDR B)
         BUF)
     ;; The char at BUF is quoted. This is why TTSKREAD failed here. Just return the list we're now inside
     (RETURN (AND (NOT NOTIFQUOTED)
                  INNERMOSTLIST)))
    (T
     (SETQ B (CDR B)))))) ; skip over escape char
(StringDelim.RC
  (COND
    ((AND (NOT ESCAPED)
         (NOT (SETQ B (FIND.MATCHING.QUOTE (CDR B)
                                           BUF))))
     ; Termination analogous to previous case
     (RETURN (AND (NOT NOTIFQUOTED)
                  INNERMOSTLIST))))
    (MULTIPLE-ESCAPE.RC
     (SETQ ESCAPED (NOT ESCAPED)))
    (OTHER.RC NIL)
    (PROGN (COND
            ((AND (EQ (CAR B)
                    (CHARCODE \;))
                 (READTABLEPROP RDTBL 'COMMONLISP))
             ; Handle semicolon special
             (COND
              ((|do| (SETQ B (CDR B))
                   (COND
                    ((EQ B BUF)
                     (RETURN T))
                    ((EQ (FIRSTCHAR B)
                       (CHARCODE EOL))
                     (RETURN)))
                   ; Done inside a comment
                   (RETURN (AND (NOT NOTIFQUOTED)
                               INNERMOSTLIST))))))))
            (SETQ B (CDR B))
            (GO LP))))))

```

(TEEXEC.OPENTEXTSTREAM

```

(LAMBDA (WINDOW MENUFN) ; Edited 13-Jun-90 00:17 by mitani
  ;; Initialize and return TEDIT TEXTSTREAM
  (LET* ((TEXTSTREAM (OPENTEXTSTREAM NIL WINDOW NIL NIL (LIST 'TERMTABLE \\PRIMTERMTABLE 'PROMPTWINDOW
                                                             'DON\'T)))
         (TEXTOBJ (TEXTOBJ TEXTSTREAM))
         (TEXTLEN (|fetch| (TEXTOBJ TEXTLEN)
                          TEXTOBJ)) ; force shift select typein to be put in keyboard buffer
         (|replace| (TEXTOBJ TXTEDITING) |of| TEXTOBJ |with| T)
         (TEXTPROP TEXTSTREAM 'STARTINGEOF TEXTLEN)
         (TEXTPROP TEXTSTREAM 'COPYBYBKSYSBUF T) ; forces COPY-SELECT to unread chars into TTY buffer
         (TEXTPROP TEXTSTREAM 'SELFN (FUNCTION \\TEEXEC.SELFN)) ; Limits selection to current input
         (|replace| (STREAM STRMBOUTFN) |of| TEXTSTREAM |with| '\\TEEXEC.TEXTBOUT)
         (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| T)
         (|replace| (SELECTION L1) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| (LIST (|fetch| DESC
                                                                                       |of| (|fetch| (TEXTOBJ THISLINE
                                                                                       )
                                                                                       |of| TEXTOBJ))))
         ; hookup middle button menu instead of TEDIT menu
         (WINDOWPROP WINDOW 'TEDIT.TITLEMENUFN MENUFN)
         (CHANGEFONT (|fetch| (CHARLOOKS CLFONT) |of| (|fetch| (TEXTOBJ CARETLOOKS) |of| TEXTOBJ)
                      TEXTSTREAM)
                    TEXTSTREAM))
  TEXTSTREAM))

```

(TEEXEC.DEFAULT.MENUFN

```

(LAMBDA (W) ; Edited 28-Mar-2024 00:06 by rmk
            ; Edited 13-Jun-90 00:16 by mitani
  ;; Default User Fn for TEEXEC windows--displays a menu of items & acts on the commands received.
  (PROG ((TEXTOBJ (TEXTOBJ W))
        (WMENU (WINDOWPROP W 'TEDIT.MENU))
        THISMENU CH OFILE OCURSOR PCTB LINES SEL ITEM)
    (COND
      ((EQ (|fetch| (TEXTOBJ EDITOPACTIVE) |of| TEXTOBJ)
           T)
       ; We're busy doing something, but not sure what. Give a general
       ; 'please wait' msg
       (TEDIT.PROMPTPRINT TEXTOBJ "Edit operation in progress; please wait." T)
      ))
  )

```

```

(RETURN))
((|fetch| (TEXTOBJ EDITOPACTIVE) |of| TEXTOBJ) ; We know specifically what's happening. Tell him
 (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (|fetch| (TEXTOBJ EDITOPACTIVE) |of| TEXTOBJ)
 " in progress; please wait."))
T)
(RETURN)))
(SETQ PCTB (|fetch| (TEXTOBJ PCTB) |of| TEXTOBJ))
(SETQ THISMENU (COND
 (WMENU)
 ((SETQ WMENU (WINDOWPROP W 'TEDIT.MENU.COMMANDS))
 (PROG1 (SETQ WMENU (TEEXEC.CREATEMENU WMENU))
 (WINDOWPROP W 'TEDIT.MENU WMENU))
 (TEEXEC.DEFAULT.MENU)))
(SETQ ITEM (MENU THISMENU))
(ERSETQ (RESETLST
 (RESETSAVE (\\TEDIT.MARKACTIVE TEXTOBJ)
 ' (AND (\\TEDIT.MARKINACTIVE OLDVALUE)))
 (|replace| (TEXTOBJ EDITOPACTIVE) |of| TEXTOBJ |with| (OR (CAR ITEM)
 T))) ; So we ca ntel the guy WHAT op is active.
(SELECTQ (CAR ITEM)
 (|Put| (TEDIT.PUT TEXTOBJ NIL NIL (TEXTPROP TEXTOBJ 'CLEARPUT)))
 (|Plain-Text| (TEDIT.PUT TEXTOBJ NIL NIL T)) ; Write out the file in the OLD TEdit format.
 (|Old-Format| (TEDIT.PUT TEXTOBJ NIL NIL NIL T))
 (|Get| (TEEXEC.GET TEXTOBJ NIL (TEXTPROP TEXTOBJ 'CLEARGET)) ; Get a new file (overwriting the one being edited.)
 (|Unformatted Get| (TEEXEC.GET TEXTOBJ NIL T))
 (|Include| (TEEXEC.INCLUDE TEXTOBJ)) ; Insert a file where the caret is
 (|ForwardFind| (TEEXEC.FIND.FORWARD TEXTOBJ)) ; Normal forward search Find
 (|BackwardFind| (TEEXEC.FIND.BACKWARD TEXTOBJ)) ; Backward search Find
 (|Limit| (SETQ TEEXEC.BUFFERLIMIT (RNUMBER)))
(COND
 ((CAR ITEM) ; This is a user-supplied entry. Get the function, and apply it to
 the TEXTSTREAM for him
 (APPLY* (CAR ITEM)
 (|fetch| (TEXTOBJ STREAMHINT) |of| TEXTOBJ)))))))))

```

(TEEXEC.DO?CMD

(LAMBDA (TEXTOBJ)

; Edited 13-Jun-90 00:16 by mitani

;;; Handles 'read macros' ? and ?=. CMD is one of those. Returns NIL if thinks it isn't. Saves current cursor location for later restoration

```

(DECLARE (SPECVARS \\?TAIL \\?PARAMS \\BUFFER)) (* (\\CARET.DOWN))
(PROG ((START (TEXTPROP TEXOBJ 'STARTINGEOF))
 (STREAM (TEXTSTREAM TEXTOBJ))
 \\BUFFER START-BUFFER FN FNSTART FNEND SPTAIL SAVE)
 (SETFILEPTR STREAM START)
 (SETQ \\BUFFER (FOR I FROM 0 TO (IDIFFERENCE (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
 START)
 COLLECT (BIN STREAM)))
 (SETQ START-BUFFER (BACKSKREAD (FLAST \\BUFFER)))
 (SELECTC (\\SYNCODE \\RDTBLSA (FIRSTCHAR START-BUFFER))
 ((LIST LEFTPAREN.RC LEFTBRACKET.RC)
 (COND
 ((AND (EQ (SCANFORWARD (CAR START-BUFFER)
 (SETQ FNSTART TTYIN-BUFFER))
 START)
 (PROGN ; START is the first paren in buffer, so check and see if there's
 an atom before it
 (SETQ FN (TTRATOM))
 )))
 )))
 )))
 (SETQ FNEND \\BUFFER)
 (EQ (TTSKIPSEPR
 START))) ; This is first list on line, preceded by FN in evalqt format
 )
 (T (SETQ FNSTART (SETQ \\BUFFER (CDR START))) ; EVAL form: read fn
 (COND
 ((EQ (SETQ FN (TTRATOM))
 CMD) ; Hasn't typed the fn name yet!
 (RETURN)))
 (SETQ FNEND \\BUFFER))))
 (PROGN ; Not inside a list now, so no macro
 (RETURN))
 (SAVE.CURSOR)

```

;;; removed: (COND ((OR (EQ FN 'E) (AND (EQ \\PROMPT1 '*') (FMEMB FN EDITCOMSL)))) ; What looks like a fn in apply format is really a command,
;;; either E or an editor command (SETQ FNSTART \\BUFFER) (SETQ FN (TTRATOM))))

```
(COND
  ((NLSETQ (PROG ((\?PARAMS |null|)
    STUFF)
    (COND
      ((EQ CMD '?')
        (XHELPSYS FN))
      (T (GO.TO.FREELINE)
        (COND
          ((EQ \BUFFER START) ; Apply format, skip over paren
            (SETQ \BUFFER (CDR START))))
          (COND
            ((OR (NOT TTYIN?=FN)
              (NOT (SETQ STUFF (APPLY* TTYIN?=FN FN))))
              (SETQ STUFF (SMARTARGLIST FN T (SETQ SPTAIL (CONS FN))))
              (COND
                ((NEQ FN (SETQ FN (CAR SPTAIL)))
                  ; Fn was spelling corrected, so There was an extra crlf involved
                  ; in printing the correction
                  (TTCLRF.ACCOUNT))
                (T (SETQ SPTAIL NIL)))
                (TTYIN.PRINTARGS FN STUFF T))
                (EQ (CAR (LISTP STUFF))
                  'ARGS)
                (TTYIN.PRINTARGS FN (CDR STUFF)
                  T))
                ((LISTP STUFF)
                  (TTPRIN2 STUFF))
                ((NEQ STUFF T)
                  (TTPRIN1 STUFF))))))))))
  ((BEEP) ; error occurred, probably undefined fn.
  ))
(SELECTQ CMD
  (? ; now delete the ?
    (TRUBOUT))
  (?= (RESTORE.CURSOR)
    (BACKWARD.DELETE.TO \?TAIL)
    (COND
      (SPTAIL
        ;; Fn was spelling corrected, so replace it. There was also an extra crlf involved in printing the correction
        (SETQ SAVE \CURSOR)
        (MOVE.TO.WHEREVER FNEND)
        (BACKWARD.DELETE.TO FNSTART)
        (READFROMBUF (CHCON FN T RDTBL))
        (MOVE.TO.WHEREVER SAVE))))
  NIL)
(RETURN T)))
```

(TEEXEC.CREATEMENU

```
(LAMBDA (ITEMS) ; (* AJB " 2-Jan-86 15:23")
  ; * [Create] \a [TEdit] [command] [menu.] [given] \a [list] [of]
  ; [menu] [items.]

  (|create| MENU
    ITEMS _ ITEMS
    CENTERFLG _ T
    MENUFONT _ (FONTCREATE 'HELVETICA 10 'BOLD)
    WHENHELDFN _ 'TEEXEC.MENU.WHENHELDFN
    WHENSELECTEDFN _ '\TEDIT.MENU.WHENSELECTEDFN))
```

(TEEXEC.GET

```
(LAMBDA (TEXTOBJ FILE UNFORMATTED?) ; Edited 30-May-91 19:17 by jds
  ; Get a file (overwriting the current textstream data)
  (PROG (OFILE OCURSOR LINES USER.CMFILE RESP TITLE FILENAME MENUSTREAM (GETFN (TEXTPROP TEXTOBJ 'GETFN))
    (SEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
    (PCTB (|fetch| (TEXTOBJ PCTB) |of| TEXTOBJ))
    (TEDIT.GET.FINISHEDFORMS NIL))
  (COND
    ((AND (|fetch| (TEXTOBJ \DIRTY) |of| TEXTOBJ)
      (PROGN (AND (|fetch| (TEXTOBJ PROMPTWINDOW) |of| TEXTOBJ)
        (FRESHLINE (|fetch| (TEXTOBJ PROMPTWINDOW) |of| TEXTOBJ)))
        (NOT (MOUSECONFIRM "Not saved yet; LEFT go Get anyway." T (|fetch| (TEXTOBJ PROMPTWINDOW)
          |of| TEXTOBJ))))))
      ; Only do the GET if he knows he'll zorch himself.
      (RETURN))
    (SETQ OFILE (OR FILE (\TEDIT.MAKEFILENAME (TEDIT.GETINPUT TEXTOBJ "File to GET: "))))
    (COND
      ((AND OFILE (INFILEP OFILE)) ; Only if there's a file to load and the file exists.
        (COND
          ((AND GETFN (EQ (APPLY* GETFN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXTOBJ)
            (FULLNAME OFILE)
            'BEFORE)
            'DON'T))
            ; He doesn't want this document put. Bail out.
            (RETURN)))
```

```
(RESETLST
  (RESETSAVE (TTYDISPLAYSTREAM (OR (|fetch| (TEXTOBJ PROMPTWINDOW) |of| TEXTOBJ
                                     PROMPTWINDOW)))
  (RESETSAVE (CURSOR WAITINGCURSOR))
  (\\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ
               NIL NIL)
  (\\TEXTCLOSEF (|fetch| (TEXTOBJ STREAMHINT) |of| TEXTOBJ)
                ; CLOSE the old files
                ; And open the new one.
  (SETQ OFILE (OPENSTREAM OFILE 'INPUT))
  (SETQ PCTB (|replace| (TEXTOBJ PCTB) |of| TEXTOBJ |with| (TEDIT.BUILD.PCTB OFILE TEXTOBJ NIL NIL
                                                             (|fetch| (TEXTOBJ DEFAULTCHARLOOKS)
                                                             |of| TEXTOBJ
                                                             (|fetch| (TEXTOBJ FMTSPEC)
                                                             |of| TEXTOBJ)
                                                             UNFORMATTED?)))
  (|for| FORM |in| TEDIT.GET.FINISHEDFORMS |do| (EVAL FORM))
                ; Do any necessary cleanup for outside packages
  (SETQ LINES (|fetch| (TEXTOBJ LINES) |of| TEXTOBJ))
  (|replace| (TEXTOBJ \\DIRTY) |of| TEXTOBJ |with| NIL)
  (|for| FIRSTLINE |inside| LINES |do| (|replace| (LINEDESCRIPTOR NEXTLINE) |of| FIRSTLINE |with| NIL))
  (|replace| (TEXTOBJ \\INSERTPCVALID) |of| TEXTOBJ |with| NIL)
  ;; The old cached piece is no longer valid--keep people from stepping on it, to prevent lost type-in and smashing other
  ;; documnts to which it has been moved...
  (TEXTPROP TEXTOBJ 'STARTINGEOF (|replace| (TEXTOBJ TEXTLEN) |of| TEXTOBJ
                                           |with| (SUB1 (\\EDITELT PCTB (SUB1 (\\EDITELT PCTB
                                                                           |\\PCTBLastPieceOffset|
                                                                           )))))
  (|replace| (SELECTION CH#) |of| SEL |with| (|replace| (SELECTION CHLIM) |of| SEL |with| TEXTLEN))
  (|replace| (SELECTION DCH) |of| SEL |with| 0)
  (|replace| (SELECTION POINT) |of| SEL |with| 'LEFT)
  (|replace| (SELECTION SET) |of| SEL |with| T)
  (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ SCRATCHSEL) |of| TEXTOBJ) |with| NIL)
  (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ) |with| NIL)
  (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ) |with| NIL)
  (|replace| (SELECTION SET) |of| TEDIT.SELECTION |with| NIL)
  (|replace| (SELECTION SET) |of| TEDIT.SHIFTEDSELECTION |with| NIL)
  (|replace| (TEXTOBJ CARETLOOKS) |of| TEXTOBJ |with| (\\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL))
  (|for| WINDOW |inside| (|fetch| (TEXTOBJ \\WINDOW) |of| TEXTOBJ) |as| LINE |inside| LINES
    |do|
      (\\FILLWINDOW (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
                  LINE TEXTOBJ NIL WINDOW))
  (\\FIXSEL SEL TEXTOBJ)
  (\\SHOWSEL SEL NIL T)
  (SETQ TITLE (TEXTSTREAM.TITLE TEXTOBJ)) ; find and set the title
  (\\TEDIT.WINDOW.TITLE TEXTOBJ (\\TEDIT.ORIGINAL.WINDOW.TITLE TITLE NIL))
  (SETQ MENUSTREAM (TEDITMENU.STREAM TEXTOBJ))
  (COND
    ((AND MENUSTREAM (|type?| LITATOM TITLE)) ; if we have a filename then put it in the GET and PUT fields of
      ; the menu
      (SETQ FILENAME (PACKFILENAME 'VERSION NIL 'BODY TITLE))
      (MBUTTON.SET.FIELD MENUSTREAM ' |Get| FILENAME)
      (MBUTTON.SET.FIELD MENUSTREAM ' |Put| FILENAME)))
    (\\TEDIT.SET.WINDOW.EXTENT TEXTOBJ (\\TEDIT.PRIMARYW TEXTOBJ))
    (\\TEDIT.HISTORYADD TEXTOBJ (|create| TEDIT.HISTORYEVENT
                                         THACTION _ ' |Get|)))
  (AND GETFN (APPLY* GETFN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXTOBJ)
                          (FULLNAME (|fetch| (TEXTOBJ TXTFILE) |of| TEXTOBJ)
                                    'AFTER)))
  (OFILE (TEDIT.PROMPTPRINT TEXTOBJ "[File not found.]"))
  (T (TEDIT.PROMPTPRINT TEXTOBJ "[Get aborted.]" T))))))
```

TEEXEC.INCLUDE

(LAMBDA (STREAM FILE START END) ; Edited 30-May-91 19:17 by jds

(* |Obtain| \\a |file| |name,| |and| |include| |that| |file's| |contents| |at| |the| |place| |where| |the| |caret| |is|.)

(* |Returns| T |if| |the| |insertion| |happened,| NIL |if| |there| |was| |no| |place| |to| |put| |it|.)

```
(SETQ STREAM (TEXTOBJ STREAM))
(PROG ((SEL (|fetch| (TEXTOBJ SEL) |of| STREAM))
      PCTB TEXTLEN NFILE NFILE INSERTCH# INSPC LEN INSPC# PCLST NPC WASOPEN PCCOUNT NSTREAM)
  (COND
    ((|fetch| (SELECTION SET) |of| SEL) (* |There| |is| \\a |place| |to| |do| |the| |include|.))
    (SETQ NFILE (OR FILE (\\TEDIT.MAKEFILENAME (TEDIT.GETINPUT STREAM "Name of the file to load: ")
                                             )
                  (COND
                    ((NOT NFILE) (* |If| |no| |file| |was| |given,| |don't| |bother| |INCLUDEing|.))
                    (TEDIT.PROMPTPRINT STREAM "[Include aborted.]" T)
                    (RETURN))
                    ((STREAMP NFILE))
                    ((NOT (INFILEP NFILE)) (* |Can't| |find| |the| |file,| |Put| |out| \\a |message|.))
                    (TEDIT.PROMPTPRINT STREAM "[File not found.]")
                    (RETURN)))
    (SETQ NFILE (OPENSTREAM '{NODIRCORE} 'OUTPUT 'NEW))
    (* |Create| |the| |holding| |file|
```

```

(SETQ NFILE (COND
  ((OPENP NFILE)
   (SETQ WASOPEN T)
   NFILE)
  (T
   (OPENFILE NFILE 'INPUT)))) (* |Wasn't| |open| -- |need| |to| |open| |it| |for| |input...|)
                                (* |And| |copy| |the| |file-section| |into| |it|.|)
(COPYBYTES NFILE NNFILE (OR START 0)
 (OR END (GETEOFPTR NFILE)))

(* |Have| |to| |explicitly| |fill| |in| 0 |and| EOFPTR\, |because| |if| |the| |file| |was| |open| |already,| NIL's |would| |only| |copy|
|from| |current| |fileptr| |to| EOF.)

(OR WASOPEN (CLOSEF NFILE)) (* |If| |the| |file| |didn't| |come| |to| |use| |open,| |close| |it|.|)
(CLOSEF NNFILE)
(SETQ NFILE NNFILE)
(SETQ START (SETQ END NIL)) (* |Then| |pretend| |nothing| |happened|.|)
(TEDIT.DO.BLUEPENDINGDELETE SEL STREAM) (* |Delete| |any| |text,| |if| |need| |be|)
(SETQ TEXTLEN (|fetch| (TEXTOBJ TEXTLEN) |of| STREAM))

(* |We| |need| |the| |POST-deletion| |text| |length| |for| |later,| |so| |this| |must| |come| |after| |the| |b-p-d|.|)

(\\SHOWSEL SEL NIL NIL) (* |Turn| |off| SEL's |before| |we| |go| |any| |further|)
(SETQ NFILE (TEXTOBJ (SETQ NSTREAM (OPENTEXTSTREAM (OPENSTREAM NFILE 'INPUT)
  NIL NIL NIL (LIST 'FONT (\\TEDIT.GET.INSERT.CHARLOOKS
  STREAM SEL)
  'PARALOOKS
  (|fetch| (TEXTOBJ FMTSPEC) |of| STREAM))))))
                                (* |Get| |a| |textobj| |to| |describe| |the| |include| |source| |file|
|need| NSTREAM |so| |that| |if| |we| |have| |to| |convert| |it| |to|
|formatted,| |we| |won't| |have| |lost| |the| |textstream--and| |thus|

|smash| |the| |free| |list|.))
(COND
  ((AND (|fetch| (TEXTOBJ FORMATTEDP) |of| NFILE)
  (NOT (|fetch| (TEXTOBJ FORMATTEDP) |of| STREAM))))
                                (* |If| |the| |included| |text| |is| |formatted| |but| |this| |file| |isn't,|
|let's| |format| |it!|)
  (\\TEDIT.CONVERT.TO.FORMATTED STREAM))
  ((AND (|fetch| (TEXTOBJ FORMATTEDP) |of| STREAM)
  (NOT (|fetch| (TEXTOBJ FORMATTEDP) |of| NFILE))))

(* |The| TARGET |document| |is| |formatted,| |but| |the| INCLUDE'd |text| |isn't,|
|Better| |format| |it| |before| |completing| |the| |include|.|)

(\\TEDIT.CONVERT.TO.FORMATTED NFILE)))
(SETQ PCTB (|fetch| (TEXTOBJ PCTB) |of| STREAM)) (* |HERE,| |because| |the| |conversion| |to| |formatted| |will|
|lengthen| |the| |pctb|)
(SETQ INSERTCH# (COND
  ((EQ (|fetch| (SELECTION POINT) |of| SEL)
  'LEFT)
  (|fetch| (SELECTION CH#) |of| SEL))
  (T (|fetch| (SELECTION CHLIM) |of| SEL)))) (* |Find| |the| |place| |to| |make| |the| |insertion|.|)
(SETQ INSPC# (OR (\\CHTOPCNO INSERTCH# PCTB)
  (\\EDITELT PCTB |\\PCTBLastPieceOffset|)))
                                (* |Likewise,| |this| |is| |affected| |by| |the| |convert-to-formatted|)
(SETQ INSPC (\\EDITELT (|fetch| (TEXTOBJ PCTB) |of| STREAM)
  (ADD1 INSPC#))) (* |The| |piece| |to| |make| |the| |insertion| |in|)
(COND
  ((NEQ INSPC 'LASTPIECE)
  (COND
    ((IGREATERP INSERTCH# (\\EDITELT PCTB INSPC#))
     (* |Must| |split| |the| |piece|.|)
     (SETQ INSPC (\\SPLITPIECE INSPC INSERTCH# STREAM INSPC#))
     (|add| INSPC# |\\EltPerPiece|)
     (SETQ PCTB (|fetch| (TEXTOBJ PCTB) |of| STREAM)))
     (* |Refresh| |the| PCTB |in| |case| |it| |grew|.|)
    ))))
(SETQ PCLST (|fetch| (TEXTOBJ PCTB) |of| NFILE)) (* |A| |temporary| |pctb,| |holding| |the| |pieces| |which| |describe|
|the| INCLUDE'd |text|)
(SETQ LEN (SUB1 (\\EDITELT PCLST (SUB1 (\\EDITELT PCLST |\\PCTBLastPieceOffset|))))))
(SETQ PCCOUNT (IDIFFERENCE (SUB1 (\\EDITELT PCLST |\\PCTBLastPieceOffset|))
  |\\FirstPieceOffset|)) (* |Remember| |how| |many| |slots| |in| |the| PCTB |we| |took|
|up| |(i.e. 2 \\x # |of| |pieces|))
(\\TEDIT.INSERT.PIECES STREAM INSERTCH# (SETQ PCLST (\\EDITELT PCLST (ADD1 |\\FirstPieceOffset|)))
  LEN INSPC INSPC# NIL)
(COND
  ((AND (|fetch| (TEXTOBJ FORMATTEDP) |of| STREAM)
  (NOT (|fetch| (TEXTOBJ FORMATTEDP) |of| NFILE))))
                                (* |If| |the| |included| |text| |is| |formatted| |but| |this| |file| |isn't,|
|let's| |format| |it!|)
  (\\TEDIT.CONVERT.TO.FORMATTED STREAM INSERTCH# (IPLUS INSERTCH# LEN))))
(\\TEDIT.HISTORYADD STREAM (|create| TEDITHISTORYEVENT
  THACTION _ ' |include|
  THCH# _ INSERTCH#
  THLEN _ LEN
  THFIRSTPIECE _ PCLST))
                                (* |Remember| |that| |we| |did| |this,| |so| |it| |can| |be| |undone|.|)

```

```
(|replace| (TEXTOBJ TEXTLEN) |of| STREAM |with| (IPLUS TEXTLEN LEN))
(* |Inserting| |the| |pieces| |didn't| |fix| |up| |things| |like| |the| |length| |of| |the| |document,| |so| |do| |it| |now.|)
(AND (|fetch| (TEXTOBJ \\WINDOW) |of| STREAM)
      (\\FIXILINES STREAM SEL INSERTCH# LEN TEXTLEN))
(|replace| (SELECTION CHLIM) |of| SEL |with| (|replace| (SELECTION CH#) |of| SEL |with| (IPLUS INSERTCH# LEN)
      ))
(* |Now| |fix| |up| |the| |selection| |to| |be| |at| |the| |end| |of| |the| |included| |text,| |point_left,| |character| |selection| |grain.|)
(|replace| (SELECTION DCH) |of| SEL |with| 0)
(|replace| (SELECTION DX) |of| SEL |with| 0)
(|replace| (SELECTION POINT) |of| SEL |with| 'LEFT) (* S'o |that| |several| |things| INCLUDED |in| |sequence| |fall| |in|
|sequence.|)
(|replace| (SELECTION SELKIND) |of| SEL |with| 'CHAR)
(|replace| (SELECTION SELOBJ) |of| SEL |with| NIL)
(COND
  ((|fetch| (TEXTOBJ \\WINDOW) |of| STREAM) (* |We're| |displaying;| |update| |the| |display| |and| |the|
|selection's| |line| |references|)
    (TEDIT.UPDATE.SCREEN STREAM)
    (\\FIXSEL SEL STREAM)
    (\\SHOWSEL SEL NIL T)))
  (|replace| (TEXTOBJ \\DIRTY) |of| STREAM |with| T) (* |Mark| |the| |document| |changed|)
  (\\SETUPGETCH (|create| EDITMARK
                PC _ INSPC
                PCOFF _ 0
                PCNO _ (IPLUS INSPC# PCCOUNT))
    STREAM) (* |Set| |the| |fileptr| |to| |the| |end| |of| |the| |insertion.|)
  (TEDIT.SCROLL? STREAM) (* |Scroll| |the| |end| |into| |view| |if| |necessary|)
  T)
(T (TEDIT.PROMPTPRINT STREAM "Please choose the place for the INCLUDE first." T))))
```

(TEEXEC.FIND.FORWARD

(LAMBDA (TEXTOBJ)

; Edited 30-May-91 19:17 by jds

(* |Case| |sensitive| |search,| |with| * |and| \\# |wildcards|)

```
(PROG* (FTEXT CHL (SEL (|fetch| (TEXTOBJ SCRATCHSEL) |of| TEXTOBJ)
  (SHIFTEDSELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)))
  (MOVESELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)))
  (DELETESELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ DELETESEL) |of| TEXTOBJ)))
  (W (|fetch| (TEXTOBJ SELWINDOW) |of| TEXTOBJ))
  (LINES (WINDOWPROP W 'LINES))
  (LINE (|fetch| (LINEDESCRIPTOR NEXTLINE) |of| LINES))
  (WREG (DSPCLIPPINGREGION NIL W))
  (WHEIGHT (|fetch| (REGION HEIGHT) |of| WREG))
  (WWIDTH (|fetch| (REGION WIDTH) |of| WREG)))
  (SETQ FTEXT (TEDIT.GETINPUT TEXTOBJ "Text to find: " (WINDOWPROP W 'TEDIT.LAST.FIND.STRING)
    (CHARCODE (EOL LF ESC))))
  (COND
    (FTEXT (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)
```

(* |f| |the| |selection| |is| |at| |the| |caret,| |then| |always| |start| |there,|
|Otherwise| |check| |if| |a| |previous| |find| |is| |in| |view,| |and| |if| |not| |just| |start| |at| |the| |caret|)

```
(SETQ CHL (TEDIT.FIND TEXTOBJ (MKSTRING FTEXT)
  (COND
    ((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
      1)
    ((WINDOWPROP W 'TEDIT.LAST.FIND.POSITION)
      (IPLUS (WINDOWPROP W 'TEDIT.LAST.FIND.POSITION)
        2))
    (T 1))
  NIL T))
(COND
  (CHL (* |We| |found| |the| |target| |text.|)
    (TEDIT.PROMPTPRINT TEXTOBJ "Done.")
    (COND
      ((|fetch| (SELECTION SET) |of| SEL) (* |Turn| |any| |selections| |off|)
        (\\SHOWSEL SEL NIL NIL)))
      (|replace| (SELECTION SET) |of| SEL |with| T)
      (|for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXTOBJ) |do| (\\EDIT.UPCARET CARET))
        (* |Remove| |any| |visible| |carets|)
      (\\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)
        NIL NIL)
      (|replace| (SELECTION CH#) |of| SEL |with| (CAR CHL))
        (* |Set| |up| SELECTION |to| |be| |the| |found| |text|)
      (|replace| (SELECTION CHLIM) |of| SEL |with| (ADD1 (CADR CHL)))
      (|replace| (SELECTION DCH) |of| SEL |with| (ADD1 (DIFFERENCE (CADR CHL)
        (CAR CHL))))
      (|replace| (SELECTION POINT) |of| SEL |with| 'RIGHT)
      (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| NIL)
      (|replace| (SELECTION SELKIND) |of| SEL |with| 'CHAR)
      (|replace| (SELECTION \\TEXTOBJ) |of| SEL |with| TEXTOBJ)
      (TEDIT.RESET.EXTEND.PENDING.DELETE SEL)
        (* |And| |never| |pending| |a| |deletion.|)
```

```

(|for| LINE |inside| (|fetch| (SELECTION L1) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
|when| LINE |do| (|replace| (LINEDESCRIPTOR YBOT) |of| LINE
|with| (SUB1 (|fetch| (REGION BOTTOM) |of| WREG)))
(* |Make| |sure| |it| |thinks| |the| |old| |selection| |is| |off-screen|
|for| |now|)
(|for| LINE |inside| (|fetch| (SELECTION LN) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
|when| LINE |do| (|replace| (LINEDESCRIPTOR YBOT) |of| LINE
|with| (SUB1 (|fetch| (REGION BOTTOM) |of| WREG)))
(BITBLT NIL 0 0 W 0 (|fetch| (REGION BOTTOM) |of| WREG)
WWIDTH
(IDIFFERENCE WHEIGHT (|fetch| (REGION BOTTOM) |of| WREG))
'TEXTURE
'REPLACE WHITESHADE)
(SETQ LINE (\\TEDIT.FIND.FIRST.LINE TEXTOBJ WHEIGHT (CAR CHL)
W))
(|replace| (LINEDESCRIPTOR YBOT) |of| LINE |with| (IDIFFERENCE WHEIGHT
(|fetch| (LINEDESCRIPTOR LHEIGHT)
|of| LINE)))
(|replace| (LINEDESCRIPTOR YBASE) |of| LINE |with| (IPLUS (|fetch| (LINEDESCRIPTOR YBOT)
|of| LINE)
(|fetch| (LINEDESCRIPTOR DESCENT)
|of| LINE)))
(\\DISPLAYLINE TEXTOBJ LINE W)
(\\FILLWINDOW (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
LINE TEXTOBJ NIL W)
(|replace| (SELECTION HASCARET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| NIL)
(|replace| (SELECTION HASCARET) |of| SEL |with| NIL)
(* |Make| |sure| |selection| |has| |no| |caret|)
(|replace| (SELECTION HOWHEIGHT) |of| SEL |with| 16384)
(* |Full| |height| |for| |highlighting|)
(COND
((|fetch| (SELECTION SET) |of| SEL)
(\\FIXSEL SEL TEXTOBJ)
(\\SHOWSEL SEL NIL T)))
(COND
((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ))
(\\FIXSEL (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)
TEXTOBJ)
(AND SHIFTEDSELWASON (\\SHOWSEL (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)
NIL T))))
(COND
((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ))
(\\FIXSEL (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)
TEXTOBJ)
(AND MOVESELWASON (\\SHOWSEL (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)
NIL T))))
(COND
((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ DELETSEL) |of| TEXTOBJ))
(\\FIXSEL (|fetch| (TEXTOBJ DELETSEL) |of| TEXTOBJ)
TEXTOBJ)
(AND DELETSELWASON (\\SHOWSEL (|fetch| (TEXTOBJ DELETSEL) |of| TEXTOBJ)
NIL T))))
(\\TEDIT.SET.WINDOW.EXTENT TEXTOBJ W)
(WINDOWPROP W 'TEDIT.LAST.FIND.STRING FTEXT)
(* |And| |get| |it| |into| |the| |window|)
(WINDOWPROP W 'TEDIT.LAST.FIND.POSITION (CAR CHL))
(* |Store| |last| |found| |position|)
)
(T (TEDIT.PROMPTPRINT TEXTOBJ "(not found)")
(WINDOWPROP W 'TEDIT.LAST.FIND.POSITION NIL)
(* |reset| |last| |position| |found|)
(\\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)
NIL T))))))

```

(TEEXEC.FIND.BACKWARD

(LAMBDA (TEXTOBJ)

; Edited 30-May-91 19:17 by jds

(* |Case| |sensitive| |search| |with| * |and| \\# |wildcards|)

```

(PROG* (FTEXT CHL (SEL (|fetch| (TEXTOBJ SCRATCHSEL) |of| TEXTOBJ))
(SHIFTEDSELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)))
(MOVESELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)))
(DELETSELWASON (|fetch| (SELECTION ONFLG) |of| (|fetch| (TEXTOBJ DELETSEL) |of| TEXTOBJ)))
(W (|fetch| (TEXTOBJ SELWINDOW) |of| TEXTOBJ))
(LINES (WINDOWPROP W 'LINES))
(LINE (|fetch| (LINEDESCRIPTOR NEXTLINE) |of| LINES))
(WREG (DSPCLIPPINGREGION NIL W))
(WHEIGHT (|fetch| (REGION HEIGHT) |of| WREG))
(WWIDTH (|fetch| (REGION WIDTH) |of| WREG)))
(SETQ FTEXT (TEDIT.GETINPUT TEXTOBJ "Text to find: " (WINDOWPROP W 'TEDIT.LAST.FIND.STRING)
(CHARCODE (EOL LF ESC))))
(COND
(FTEXT (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)

```

(* |f| |the| |selection| |is| |at| |the| |caret|, |then| |always| |start| |there|. |Otherwise| |check| |if| |a| |previous| |find| |is| |in| |view|, |and| |if| |not| |just| |start| |at| |the| |caret|)

(SETQ CHL (TEDIT.FIND.BACKWARD TEXTOBJ (MKSTRING FTEXT)


```

(COND
  ((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
   (|fetch| (SELECTION CH#) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)))
  (T (OR (AND (|fetch| (SELECTION SET) |of| SEL)
             (WINDOWPROP W 'TEDIT.LAST.FIND.POSITION))
        (|fetch| (SELECTION CH#) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))))))
1 T))
(COND
  (CHL
    (* W|e |found| |the| |target| |text|.|)
    (TEDIT.PROMPTPRINT TEXTOBJ "Done.")
    (COND
      ((|fetch| (SELECTION SET) |of| SEL) (* |Turn| |any| |selections| |off|))
      (\\SHOWSEL SEL NIL NIL))
      (|replace| (SELECTION SET) |of| SEL |with| T)
      (|for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXTOBJ) |do| (\\EDIT.UPCARET CARET))
      (* |Remove| |any| |visible| |carets|)
      (\\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)
        NIL NIL)
      (|replace| (SELECTION CH#) |of| SEL |with| (SUB1 (IDIFFERENCE (CAR CHL)
        (IDIFFERENCE (CADR CHL)
          (CAR CHL))))))
      (* |Set| |up| SELECTION |to| |be| |the| |found| |text|)
      (|replace| (SELECTION CHLIM) |of| SEL |with| (CAR CHL))
      (|replace| (SELECTION DCH) |of| SEL |with| (ADD1 (IDIFFERENCE (CADR CHL)
        (CAR CHL))))))
      (|replace| (SELECTION POINT) |of| SEL |with| 'LEFT)
      (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| NIL)
      (|replace| (SELECTION SELKIND) |of| SEL |with| 'CHAR)
      (|replace| (SELECTION \\TEXTOBJ) |of| SEL |with| TEXTOBJ)
      (TEDIT.RESET.EXTEND.PENDING.DELETE SEL)
      (* |And| |never| |pending| |a| |deletion|.|)
      (|for| LINE |inside| (|fetch| (SELECTION L1) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
        |when| LINE |do| (|replace| (LINEDESCRIPTOR YBOT) |of| LINE
          |with| (SUB1 (|fetch| (REGION BOTTOM) |of| WREG))))
          (* |Make| |sure| |it| |thinks| |the| |old| |selection| |is| |off-screen|
            |for| |now|)
      (|for| LINE |inside| (|fetch| (SELECTION LN) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ))
        |when| LINE |do| (|replace| (LINEDESCRIPTOR YBOT) |of| LINE
          |with| (SUB1 (|fetch| (REGION BOTTOM) |of| WREG))))
      (BITBLT NIL 0 0 W 0 (|fetch| (REGION BOTTOM) |of| WREG)
        WWIDTH
        (IDIFFERENCE WHEIGHT (|fetch| (REGION BOTTOM) |of| WREG))
        'TEXTURE
        'REPLACE WHITESHADE)
      (SETQ LINE (\\TEDIT.FIND.FIRST.LINE TEXTOBJ WHEIGHT (CAR CHL)
        W))
      (* |Find| |the| |first| |line| |to| |go| |in| |the| |window|)
      (|replace| (LINEDESCRIPTOR YBOT) |of| LINE |with| (IDIFFERENCE WHEIGHT
        (|fetch| (LINEDESCRIPTOR LHEIGHT)
          |of| LINE)))
      (|replace| (LINEDESCRIPTOR YBASE) |of| LINE |with| (IPLUS (|fetch| (LINEDESCRIPTOR YBOT)
        |of| LINE)
        (|fetch| (LINEDESCRIPTOR DESCENT)
          |of| LINE)))
      (\\DISPLAYLINE TEXTOBJ LINE W)
      (\\FILLWINDOW (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
        LINE TEXTOBJ NIL W)
      (|replace| (SELECTION HASCARET) |of| (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ) |with| NIL)
      (|replace| (SELECTION HASCARET) |of| SEL |with| NIL)
      (* |Make| |sure| |selection| |has| |no| |caret|)
      (|replace| (SELECTION HOWHEIGHT) |of| SEL |with| 16384)
      (* |Full| |height| |for| |highlighting|)
    (COND
      ((|fetch| (SELECTION SET) |of| SEL)
       (\\FIXSEL SEL TEXTOBJ)
       (\\SHOWSEL SEL NIL T)))
    (COND
      ((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ))
       (\\FIXSEL (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)
        TEXTOBJ)
       (AND SHIFTEDSELWASON (\\SHOWSEL (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXTOBJ)
        NIL T))))
    (COND
      ((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ))
       (\\FIXSEL (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)
        TEXTOBJ)
       (AND MOVESELWASON (\\SHOWSEL (|fetch| (TEXTOBJ MOVESEL) |of| TEXTOBJ)
        NIL T))))
    (COND
      ((|fetch| (SELECTION SET) |of| (|fetch| (TEXTOBJ DELETESSEL) |of| TEXTOBJ))
       (\\FIXSEL (|fetch| (TEXTOBJ DELETESSEL) |of| TEXTOBJ)
        TEXTOBJ)
       (AND DELETESSELWASON (\\SHOWSEL (|fetch| (TEXTOBJ DELETESSEL) |of| TEXTOBJ)
        NIL T))))
      (\\TEDIT.SET.WINDOW.EXTENT TEXTOBJ W)
      (WINDOWPROP W 'TEDIT.LAST.FIND.STRING FTEXT)
      (* |And| |get| |it| |into| |the| |window|)
      (WINDOWPROP W 'TEDIT.LAST.FIND.POSITION (CAR CHL))

```

```

)
(T (TEDIT.PROMPTPRINT TEXTOBJ "(not found)")
(WINDOWPROP W 'TEDIT.LAST.FIND.POSITION NIL)
(* |reset| |last| |position| |found|)
(\\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)
NIL T))))))

```

(TEEXEC.MENU.WHENHELDFN

```

(LAMBDA (ITEM MENU BUTTON) (* AJB "30-Jan-86 13:09")
(PROMPTPRINT (SELECTQ (CAR ITEM)
(|Put| "Sends the document to a file")
(|Get| "Gets a new file as the document to edit.")
(|Include| "Includes a file at the cursor")
(|ForwardFind|
"Searches forward for a string of text")
(|BackwardFind|
"Searches backward for a string of text")
(|Limit| "Limits the number of characters in the textstream in memory")
""))))

```

(TEEXEC.SHRIK.ICONCREATE

```

(LAMBDA (W ICON ICONW) ; Edited 28-Mar-2024 00:08 by rmk
(* AJB " 7-Jan-86 16:37")
; Create the icon that represents this window.

(CL:WHEN (TEXTSTREAM W T)
(LET ((ICON (WINDOWPROP W 'ICON))
(ICONTITLE (WINDOWPROP W 'TEDIT.ICON.TITLE))
(SHRINKFN (WINDOWPROP W 'SHRINKFN)))
(COND
((WINDOWPROP W 'TEDITMENU) ; This is a text menu, and shrinks without trace.
)
((OR (IGREATERP (FLENGTH SHRINKFN)
3)
(AND (NOT (FMEMB 'SHRINKATTACHEDWINDOWS SHRINKFN))
(IGREATERP (FLENGTH SHRINKFN)
2))) ; There are other functions that expect to handle this. Don't
; bother.
)
((OR (AND ICONTITLE (EQUAL ICONTITLE (PROCESSPROP (WINDOWPROP W 'PROCESS)
'NAME)))
(AND (NOT ICONTITLE)
ICON)) ; we built this and the title is the same, or he has already put an
; icon on this. Do nothing
)
)
(ICON ; There's an existing icon window; change the title in it
(WINDOWPROP W 'TEDIT.ICON.TITLE (SETQ ICONTITLE (PROCESSPROP (WINDOWPROP W 'PROCESS)
'NAME)))
(ICONTITLE ICONTITLE NIL NIL ICON))
(T ; install a new icon
(WINDOWPROP W 'TEDIT.ICON.TITLE (SETQ ICONTITLE (PROCESSPROP (WINDOWPROP W 'PROCESS)
'NAME)))
(WINDOWPROP W 'ICON (TITLEDICONW TEEXEC.TITLED.ICON.TEMPLATE ICONTITLE TEEXEC.ICON.FONT NIL T
'(BOTTOM LEFT))))))
(WINDOWPROP W 'ICON)))

```

(TEEXEC.FILLBUFFER

```

(LAMBDA (FILLTYPE) ; Edited 30-May-91 19:18 by jds

```

;;; While filling the line, the current file pointer is the end of the line. When the line is closed, this is made the eof. --- #CURRENTRDITBL# is used for syntactic delimiters and paren counting on READ and RATOM calls but isn't referenced (or bound) for READC

```

(DECLARE (USEDFFREE \\#CURRENTRDITBL# \\PRIMTERMTABLE *READ-NEWLINE-SUPPRESS* \\TERM.OFD)
(SPECVARS RSNX TCLASS RTBLSA RAISEDCHAR FILLTYPE RAISEDCHAR PEEKEDCHOED C)
(\\RESETLINE)

```

;;; If ERROR or RESET, move STARTINGEOF to end of text (TEXTLEN)

```

(RESETLST
(RESETSAVE NIL (LIST (FUNCTION (LAMBDA NIL
(COND
(RESETSTATE ; Point to end of text and clearout linebuffer on RESET or
; ERROR
(PROG* ((TEXTOBJ (|fetch| (TEXTSTREAM TEXTOBJ) |of| \\TERM.OFD))
(SEL (|fetch| (TEXTOBJ SEL) |of| TEXTOBJ)))
(TEXTPROP TEXTOBJ 'STARTINGEOF (|fetch| (TEXTOBJ TEXTLEN)
|of| TEXTOBJ))
(|replace| (STREAM REVALIDATEFLG) |of| \\LINEBUF.OFD
|with| NIL)
(\\RESETTERMINAL)
(\\SHOWSEL SEL)

```

```
(|replace| (SELECTION CH#) |of| SEL
|with| (ADD1 (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)))
(|replace| (SELECTION CHLIM) |of| SEL
|with| (ADD1 (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)))
(|replace| (SELECTION POINT) |of| SEL |with| 'LEFT)
(|replace| (SELECTION DCH) |of| SEL |with| 0)
(|replace| (SELECTION SET) |of| SEL |with| T)
(\\FIXSEL SEL TEXOBJ))))))
```

```
(PROG* ((RTBLSA (|fetch| READSA |of| \\#CURRENTRTD#)
(CONTROLTON (|fetch| CONTROLFLG |of| \\PRINTERM#)
(TEXOBJ (|fetch| (TEXTSTREAM TEXOBJ) |of| \\TERM.OFD))
(SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
(WINDOW (|fetch| (TEXTOBJ \\WINDOW) |of| TEXOBJ))
(LINES (|fetch| (TEXTOBJ LINES) |of| TEXOBJ))
RSNX TCLASS C RAISEDCHAR PEEKEDCHOED TTYWINDOW FN TCH INSCH# CHNO ADDEDEOL)
```

;;; STARTINGEOF is the beginning of the current text being entered which gets returned to READ so that \\TEEXEC.TEXTBOUT knows where to output
 ;;; any text including ^T

;;; TCLASS is terminal syntax class, RSNX is read-table code

```
(TEXTPROP TEXOBJ 'STARTINGEOF (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
; Keep STARTINGEOF in sync
(COND
((SETQ C (|fetch| (LINEBUFFER PEEKEDCHAR) |of| \\LINEBUF.OFD))
; Account for peeked character
; The peeked char may be negative because it was BIN'ed
; earlier. Make sure it is positive.
(|replace| (LINEBUFFER PEEKEDCHAR) |of| \\LINEBUF.OFD |with| NIL)
(SETQ PEEKEDCHOED T)
(SETQ RAISEDCHAR (\\RAISECHAR C))
(COND
((EQ FILLTYPE READC.FT)
(TEXTPROP TEXOBJ 'STARTINGEOF (SUB1 (TEXTPROP TEXOBJ 'STARTINGEOF))))
; Backup one in textstream to start the input before the peeked
; and echoed character if doing a READ
))
(COND
((AND CONTROLTON (EQ FILLTYPE READC.FT))
(TEEXEC.INSERTCHAR TEXOBJ C) ; Read single char and check for echoing
(GO EXIT))) ; If in CONTROL T mode and reading a single char
(COND
(C ; Working on the previously-peeked char, so skip the keyboard read. Since the peeked char has been inserted in the TEdit
; buffer already, back up the starting-eof counter to include it.
(TEXTPROP TEXOBJ 'STARTINGEOF (SUB1 (TEXTPROP TEXOBJ 'STARTINGEOF)))
; Then go skip the kbd read.
(GO NEXTTCLASS)))
; ---
; Top of the next-character loop
; ---
(\\SHOWSEL SEL NIL NIL)
(\\SHOWSEL SEL NIL T)
NEXT
(SETQ C (TEEXEC.GETKEY TEXOBJ)) ; read next character from keyboard
NEXTTCLASS
(SETQ TCLASS (|fetch| TERMCLASS |of| (\\SYNCODE \\PRINTERMSA (SETQ RAISEDCHAR (\\RAISECHAR C))))))
REDO
```

;;; Handle Terminal Class characters

```
(SELECTQ (TEEXEC.FILLBUFFER.TCLASS TEXOBJ SEL)
(NEXT (GO NEXT))
(EXIT (GO EXIT))
NIL)
```

;;; Here if it isn't a terminal class.

```
(COND
(PEEKEDCHOED (SETQ PEEKEDCHOED NIL))
(T (TEEXEC.INSERTCHAR TEXOBJ C)))
(AND (EQ FILLTYPE READC.FT)
(GO NEXT))
(COND
((EQ ESCAPE.RC (SETQ RSNX (\\SYNCODE RTBLSA RAISEDCHAR)))
(COND
((EQ CTRLV.TC (SETQ TCLASS (|fetch| TERMCLASS |of| (\\SYNCODE \\PRINTERMSA (SETQ RAISEDCHAR
(TEEXEC.INSERTCHAR
TEXOBJ))))))
(GO REDO)))
(GO NEXT)))
(SELECTC FILLTYPE
```

```

(RATOM/RSTRING.FT
  (COND
    ((AND CONTROLTON (|fetch| STOPATOM |of| RSNX))
      (GO EXIT)))
(READ.FT (SELECTC RSNX
  ((LIST RIGHTPAREN.RC RIGHTBRACKET.RC LEFTPAREN.RC LEFTBRACKET.RC STRINGDELIM.RC
    )
  (COND
    ((NOT (TEXEC.EOTP TEXOBJ)
      (|replace| (STREAM REVALIDATEFLG) |of| \\LINEBUF.OFD |with| T)
        ; Inserting a paren/bracket in the middle of the input, invalidate
        ; paren/bracket count
      )
    )
  (SELECTC RSNX
    ((LIST RIGHTPAREN.RC RIGHTBRACKET.RC)
      (TEXEC.FLASHCARET TEXOBJ (TEXEC.PARENMATCH TEXOBJ RSNX)))
    NIL))
  NIL)
(COND
  ((AND CONTROLTON (ZEROP (|fetch| (LINEBUFFER LBRKCOUNT) |of| \\LINEBUF.OFD))
    (ZEROP (|fetch| (LINEBUFFER LPARCOUNT) |of| \\LINEBUF.OFD))
    (|fetch| STOPATOM |of| RSNX)
    (SELECTC RSNX
      ((LIST LEFTPAREN.RC LEFTBRACKET.RC RIGHTBRACKET.RC RIGHTPAREN.RC)
        NIL)
      (STRINGDELIM.RC
        (COND
          ((|fetch| (LINEBUFFER INSTRINGP) |of| \\LINEBUF.OFD)
            (|replace| (LINEBUFFER INSTRINGP) |of| \\LINEBUF.OFD |with| NIL)
            T)))
          (NOT (|fetch| (LINEBUFFER INSTRINGP) |of| \\LINEBUF.OFD))))
    ;; READ is reading an atom. Return when atom ends, but also obey bracket/paren exception noted on page
    ;; 14.33 of manual.
    (GO EXIT)))
(COND
  ((TEXEC.EOTP TEXOBJ)
    (COND
      ((|fetch| (STREAM REVALIDATEFLG) |of| \\LINEBUF.OFD)
        (TEXEC.PARENCOUNT TEXOBJ) ; text needs recount of parens/brackets
      )
    )
  (COND
    ((\\INCPARENCOUNT RSNX)
      (COND
        ((STREAMPROP \\TERM.OFD 'FIXFLG)
          (STREAMPROP \\TERM.OFD 'FIXFLG NIL)
            ; Expression is being FIXed by user, reset FIXFLG and go for
            ; more input
          (GO NEXT)))
        ;; Parens balance--throw the carriage if the closing paren or bracket character was not a CR, and if
        ;; FLG argument of READ is NIL. (We know we are under a READ call because of FILLTYPE)
        ;; copy the chars from the textstream into the linebuffer
        (TEXEC.TEXTSTREAM.TO.LINEBUF TEXOBJ (TEXTPROP TEXOBJ 'STARTINGEOF)
          \\LINEBUF.OFD FILLTYPE)
          ; AND (EQ FILLTYPE READ.FT) (TEXEC.FIX? TEXOBJ
          ; \\LINEBUF.OFD) (GO NEXT)
          ; If it was a PA FIX command handle it, and allow editing
          ; now reset the new STARTINGEOF to start at the end of the text
        (TEXTPROP TEXOBJ 'STARTINGEOF (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
        (\\CLOSELINE)
        (AND (NEQ RAISEDCHAR (CHARCODE EOL))
          (NOT *READ-NEWLINE-SUPPRESS*)
          (\\OUTCHAR \\TERM.OFD (CHARCODE EOL)))
          ; \\CLOSELINE first so dribble happens before EOL
        (RETURN))
        (EQ IMMEDIATE.RMW (|fetch| WAKEUP |of| RSNX))
          ; Immediate read-macro
        (RETURN))))))
  (SHOULDNT))
(GO NEXT)
EXIT
(COND
  ((STREAMPROP \\TERM.OFD 'FIXFLG)
    (STREAMPROP \\TERM.OFD 'FIXFLG NIL)
      (COND
        ((NEQ RAISEDCHAR (CHARCODE EOL))
          ; Expression is being FIXed by user, reset FIXFLG and go for
          ; more input
        (GO NEXT))))))
(COND
  ((AND (EQ FILLTYPE READ.FT)
    (EQ RAISEDCHAR (CHARCODE EOL))
    (EQ (SUB1 (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
      (TEXTPROP TEXOBJ 'STARTINGEOF)))
    (\\LINEBUFABOUT \\LINEBUF.OFD (CAR (GETSYNTAX 'RIGHTBRACKET \\#CURRENTRDTBL#))))

```

;; If doing a READ, force a lone CR to terminate the READ by handing back a RIGHTBRACKET into the LINEBUFFER

```

)
(T (TEXEC.TEXTSTREAM.TO.LINEBUF TEXOBJ (TEXTPROP TEXOBJ 'STARTINGEOF)
  \\\LINEBUF.OFD FILLTYPE)))
(TEXTPROP TEXOBJ 'STARTINGEOF (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
; AND (EQ FILLTYPE READ.FT) (TEXEC.FIX? TEXOBJ)
; \\\LINEBUF.OFD) (GO NEXT)
; If it was a PA FIX command handle it, and allow editing

(\\CLOSELINE)
(AND (NEQ RAISEDCHAR (CHARCODE EOL))
      (NOT CONTROLTON)
      (NOT *READ-NEWLINE-SUPPRESS*)
      (\\OUTCHAR \\TERM.OFD (CHARCODE EOL))))))

```

(TEXEC.FILLBUFFER.TCLASS

(LAMBDA (TEXOBJ SEL)

; Edited 30-May-91 19:18 by jds

;; Handle special terminal class characters

```

(DECLARE (USEDFREE \\\LINEBUF.OFD PEEKEDCHOED C FILLTYPE TCLASS))
(PROG NIL

```

```

  (SELECTC TCLASS
    (RETYPE.TC (RETURN 'NEXT) ; Ignore ^R since the user can rescroll the line

```

```

    )
    (CHARDELETE.TC
      (TEXEC.FILLBUFFER.CHARDELETE TEXOBJ)
      (RETURN 'NEXT))
    (WORDDELETE.TC
      (TEXEC.FILLBUFFER.WORDDELETE TEXOBJ)
      (RETURN 'NEXT))
    (LINEDELETE.TC
      (TEXEC.FILLBUFFER.LINEDELETE TEXOBJ)
      (RETURN 'NEXT))

```

```

    (CTRLV.TC (COND
      (PEEKEDCHOED (SETQ PEEKEDCHOED NIL)))
      (TEXEC.INSERTCHAR TEXOBJ C)
      (COND
        ((NEQ FILLTYPE READC.FT)
         (SETQ C (TEXEC.GETKEY TEXOBJ))))
      (SETQ TCLASS NONE.TC))
    (EOL.TC (AND (EQ FILLTYPE READ.FT)
      (TEXEC.?EQUAL TEXOBJ)
      (TRUE (TEXEC.INSERTCHAR TEXOBJ (CHARCODE EOL)))
      (RETURN 'EXIT) ; Let READ handle ? and ?=

```

```

    )
    (TEXEC.INSERTCHAR TEXOBJ C)
    (COND

```

((EQ FILLTYPE READ.FT) ; If we are doing a 'READ'

;; If we are at the end of the input now, update parencount if invalid, and test for matching paren/bracket count

```

    (AND (TEXEC.EOTP TEXOBJ)
      (COND
        ((|fetch| (STREAM REVALIDATEFLG) |of| \\\LINEBUF.OFD)
         (TEXEC.PARENCOUNT TEXOBJ))
        (T T))
        (ZEROP (|fetch| (LINEBUFFER LBRKCOUNT) |of| \\\LINEBUF.OFD))
        (ZEROP (|fetch| (LINEBUFFER LPARCOUNT) |of| \\\LINEBUF.OFD))
        (NOT (|fetch| (LINEBUFFER INSTRINGP) |of| \\\LINEBUF.OFD))
        (RETURN 'EXIT) ; Proper termination of READ

```

) ; Else go for more input

(T (RETURN 'EXIT) ; Exit always if not in 'READ' mode

)))

```

NIL)
(COND
  ((EQ C 127) ; DELETE

```

```

    (COND
      ((AND (|fetch| (TEXTOBJ BLUEPENDINGDELETE) |of| TEXOBJ)
            (IGREATERP (|fetch| (SELECTION CH#) |of| SEL)
                       (TEXTPROP TEXOBJ 'STARTINGEOF))) ; Only allow deletion of selection if left side of selection is after
              ; start of current input
              (\\TEDIT.DELETE SEL TEXOBJ) ; Erase characters from the screen
              (TEDIT.RESET.EXTEND.PENDING.DELETE SEL) ; Make it a normal selection again.
              (|replace| (STREAM REVALIDATEFLG) |of| \\\LINEBUF.OFD |with| T))) ; invalidate paren/bracket count
      (RETURN 'NEXT)))

```

```

(COND
  ((FMEMB C (LIST (CHARCODE ^U)
                  (CHARCODE ^Q))) ; Delete a line
    (TEXEC.FILLBUFFER.LINEDELETE TEXOBJ)

```

```

(RETURN 'NEXT)))
(COND
  (EQ C (CHARCODE ^X)) ; ^X positions to end of text
  (\SETUPGETCH (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ
    TEXTOBJ)
  (|replace| (SELECTION CH#) |of| SEL |with| (ADD1 (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)))
  (|replace| (SELECTION CHLIM) |of| SEL |with| (|fetch| (SELECTION CH#) |of| SEL))
  (|replace| (SELECTION POINT) |of| SEL |with| 'LEFT)
  (|replace| (SELECTION DCH) |of| SEL |with| 0)
  (|replace| (SELECTION SET) |of| SEL |with| T)
  (|replace| (SELECTION HASCARET) |of| SEL |with| T)
  (UNINTERRUPTABLY
    (\CHECKCARET)
    (|for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXOBJ) |as| WIN |inside| (|fetch| (TEXTOBJ \WINDOW)
      |of| TEXOBJ)
    |do| (\FIXSEL SEL TEXOBJ WIN)
      (MOVETO (|fetch| (SELECTION X0) |of| SEL)
        (|fetch| (SELECTION Y0) |of| SEL)
        WIN)
      (|replace| TCCARETX |of| CARET |with| (|fetch| (SELECTION X0) |of| SEL))
      (|replace| TCCARETY |of| CARET |with| (|fetch| (SELECTION Y0) |of| SEL)))
  (RETURN 'NEXT))))))

```

TEEXEC.CHSELPENDING

```

(LAMBDA (TEXOBJ) ; Edited 30-May-91 19:18 by jds
  (ERSETQ (PROGN (|replace| (TEXTOBJ EDITOPACTIVE) |of| TEXOBJ |with| T)
    (* |Before| |starting| |to| |work,| |note| |that| |we| |are| |doing|
      |something|)
    (COND
      (TEDIT.COPY.PENDING (* |Have| |to| |copy| |the| |shifted| SEL |to| |caret.|)
        (SETQ TEDIT.COPY.PENDING NIL)
        (\COPYSEL TEDIT.SHIFTEDSELECTION (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ))
        (TEDIT.COPY (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)
          (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
        (|replace| (SELECTION SET) |of| TEDIT.SHIFTEDSELECTION |with| NIL)
        (|replace| (SELECTION L1) |of| TEDIT.SHIFTEDSELECTION |with| NIL)
        (|replace| (SELECTION LN) |of| TEDIT.SHIFTEDSELECTION |with| NIL)
        (\COPYSEL TEDIT.SHIFTEDSELECTION (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)))
      (TEDIT.COPYLOOKS.PENDING (* |Have| |to| |copy| |the| |shifted| SEL |to| |caret.|)
        (SETQ TEDIT.COPYLOOKS.PENDING NIL)
        (\COPYSEL TEDIT.COPYLOOKSSELECTION (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ))
        (COND
          ((EQ 'PARA (|fetch| (SELECTION SELKIND) |of| (|fetch| (TEXTOBJ SHIFTEDSEL)
            |of| TEXOBJ)))
            (* |copy| |the| |paragraph| |looks,| |since| |the| |source| |selection|
              |type| |was| |paragraph|)
            (TEDIT.COPY.PARALOOKS TEXOBJ (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)
              (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)))
          (T
            (TEDIT.COPY.LOOKS TEXOBJ (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)
              (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)))
            (\SHOWSEL (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)
              NIL NIL)
            (|replace| (SELECTION SET) |of| TEDIT.COPYLOOKSSELECTION |with| NIL)
            (|replace| (SELECTION L1) |of| TEDIT.COPYLOOKSSELECTION |with| NIL)
            (|replace| (SELECTION LN) |of| TEDIT.COPYLOOKSSELECTION |with| NIL)
            (\COPYSEL TEDIT.COPYLOOKSSELECTION (|fetch| (TEXTOBJ SHIFTEDSEL) |of| TEXOBJ)))
          (TEDIT.MOVE.PENDING (* |Have| |to| |move| |the| |ctrl-shift| SEL |to| |caret.|)
            (SETQ TEDIT.MOVE.PENDING NIL)
            (\COPYSEL TEDIT.MOVESELECTION (|fetch| (TEXTOBJ MOVESEL) |of| TEXOBJ))
            (TEDIT.DO.BLUEPENDINGDELETE SEL TEXOBJ)
            (TEDIT.MOVE (|fetch| (TEXTOBJ MOVESEL) |of| TEXOBJ)
              (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
            (|replace| (SELECTION SET) |of| TEDIT.MOVESELECTION |with| NIL)
            (|replace| (SELECTION L1) |of| TEDIT.MOVESELECTION |with| NIL)
            (|replace| (SELECTION LN) |of| TEDIT.MOVESELECTION |with| NIL)
            (\COPYSEL TEDIT.MOVESELECTION (|fetch| (TEXTOBJ MOVESEL) |of| TEXOBJ)))
          (TEDIT.DEL.PENDING (* |Delete| |the| |current| |selection.|)
            (SETQ TEDIT.DEL.PENDING NIL)
            (* |Above| |all,| |reset| |the| |demand| |flag| |first|)
            (COND
              ((|fetch| (SELECTION SET) |of| TEDIT.DELETESELECTION)
                (* |Only| |try| |the| |deletion| |if| |he| |really| |set| |the| |selection.|)
                (\SHOWSEL (|fetch| (TEXTOBJ DELETEDSEL) |of| TEXOBJ)
                  NIL NIL)
                (* |Turn| |off| |the| |selection| |highlights|)
                (\SHOWSEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
                  NIL NIL)
                (|replace| (SELECTION SET) |of| (|fetch| (TEXTOBJ DELETEDSEL) |of| TEXOBJ) |with| NIL)
                (\COPYSEL TEDIT.DELETESELECTION (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
                (\TEDIT.SET.SEL.LOOKS (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
                  'NORMAL)
                (* |Grab| |the| |selection| |we're| |to| |use|)
                (\TEDIT.DELETE (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
                  (|fetch| (SELECTION \TEXTOBJ) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
                  NIL)
                (|replace| (SELECTION L1) |of| TEDIT.DELETESELECTION |with| NIL)
                (|replace| (SELECTION LN) |of| TEDIT.DELETESELECTION |with| NIL))))))
            (|replace| (TEXTOBJ EDITOPACTIVE) |of| TEXOBJ |with| NIL))))))

```

(TEXEC.FILLBUFFER.CHARDELETE

(* A.JB " 7-Jan-86 15:03")

```
(LAMBDA (TEXOBJ)
  (DECLARE (USEDFREE FILLTYPE RSNX TCLASS RTBLSA RAISEDCHAR \#CURRENTRTBL#))
  (PROG (C)
    (COND
      ((NULL (SETQ C (TEXEC.\CHDEL1 TEXOBJ)))
        (* |Try| |deleting| \a |character| (|put| |the| |deleted| |char| |in| C |and| RETURN C).\
        |f |there| |was| |no| |character| |to| |delete| (|ie.| |we're| |at| |start| |of| |line|).\
        (RETURN NIL))
        (FLASHWINDOW)
        (RETURN)))
      (PROG (C1 (ESCAPE? (AND (NEQ FILLTYPE READC.FT)
        (|fetch| ESCAPEFLG |of| \#CURRENTRTBL#)
        ESCAPE.RC)))
        (COND
          ((NEQ FILLTYPE READC.FT) (* |Don't| |process| |escapes| |if| READC)
            (SETQ RSNX (\SYNCODE RTBLSA C))
            (COND
              ((SETQ C1 (TEXEC.NTHBUFCHARBACK TEXOBJ 0))
                (* |Check| |preceding| |char| C1 |for| |escape|)
                (COND
                  ((EQ ESCAPE? (\SYNCODE RTBLSA C1))
                    (SETQ RSNX OTHER.RC)
                    (SETQ C (TEXEC.\CHDEL1 TEXOBJ)) (* |Delete| |the| ESCAPE |char| |also|)
                    )))
              (COND
                ((NULL (|fetch| (STREAM REVALIDATEFLG) |of| \LINEBUF.OFD))
                  (\DECPARENCOUNT RSNX) (* |no| |need| |to| |update| |parencount| |if| |deleting| |chars| |in|
                  |the| |middle| |of| |the| |text|)
                  )))
                (RETURN C) (* |Successful| |delete|)
              )))
    )))
```

(TEXEC.FILLBUFFER.WORDDELETE

; Edited 30-May-91 19:18 by jds

```
(* |Delete| |chars| |until| |first| |non-sepr/non-other.| |or| |first| |non-other| |after| |sepr| |string| |and| |other| |string.|
|Note| |that| \a |terminal| |wordsepr| |is| |treated| |as| |if| |it| |were| \a |read-sepr|)

(DECLARE (USEDFREE RSNX RTBLSA \PRIMTERMSA TCLASS FILLTYPE \#CURRENTRTBL# \LINEBUF.OFD))
(PROG* (C (STARTINGEOF (TEXTPROP TEXOBJ 'STARTINGEOF))
  (SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
  (ENDCH# (SUB1 (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (COND
      ((ZEROP (|fetch| (SELECTION DCH) |of| SEL))
        (|fetch| (SELECTION CHLIM) |of| SEL))
        (T (SUB1 (|fetch| (SELECTION CHLIM) |of| SEL))))))
    (RIGHT (|fetch| (SELECTION CHLIM) |of| SEL))
    NIL)))
  (CH# ENDCH#))
  (COND
    ((ILEQ CH# (TEXTPROP (CAR (|fetch| (TEXTOBJ \WINDOW) |of| TEXOBJ))
      'STARTINGEOF))
      (FLASHWINDOW)
      (RETURN)))
    (AGAIN
      (COND
        ((SETQ C (PROGN (\SETUPGETCH CH# TEXOBJ)
          (\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))))
          (SETQ RSNX (\SYNCODE RTBLSA C))
          (SELECTC RSNX
            (SEPRCHAR.RC (|add| CH# -1)
              (GO AGAIN) (* |cont| |until| |first| |non-sepr|)
              )
            (OTHER.RC (COND
              ((EQ WORDSEPR.TC (\SYNCODE \PRIMTERMSA C))
                (|add| CH# -1)
                (GO AGAIN) (* |cont| |until| |first| |non-sepr|)
                )))
              (PROG (C1 (ESCAPE? (AND (NEQ FILLTYPE READC.FT)
                (|fetch| ESCAPEFLG |of| \#CURRENTRTBL#)
                ESCAPE.RC)))
                (|The| |first| OTHER)
                (TRY (COND
                  ((SETQ C (PROGN (\SETUPGETCH CH# TEXOBJ)
                    (\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))))
                    (* |look| |at| |previous| |char|)
                    (SETQ RSNX (\SYNCODE RTBLSA C))
                    (SETQ TCLASS (\SYNCODE \PRIMTERMSA C))
                    (COND
                      ((SETQ C1 (PROGN (\SETUPGETCH (SUB1 CH#)
                        TEXOBJ)
```

```

((\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ)))
(COND
  ((EQ ESCAPE? (\SYNCODE RTBLSA C1))
   (SETQ RSNX OTHER.RC)
   (SETQ TCLASS NONE.TC))))
(COND
  ((AND (NEQ TCLASS WORDSEPR.TC)
        (EQ OTHER.RC RSNX))
   (|add| CH# -1) (* |Erase| |it|)
   (GO TRY))))))
(PROGN (|add| CH# -1)
  (COND
    ((EQ FILLTYPE READ.FT)
     (* |For| READ |filltype| |only|, |Fix| |up| |paren| |count| |if| |we| |still| |have| \a |valid| |paren| |count| |since| |this| |can| |be| \a
      |paren| |or| |bracket| |or| |stringdelimit|))
     (COND
       ((NOT (|fetch| (STREAM REVALIDATEFLG) |of| \LINEBUF.OFD))
        (\DECPARENCOUNT RSNX))))))
(TEDIT.RESET.EXTEND.PENDING.DELETE SEL) (* |Make| |it| \a |normal| |selection| |again|.))
(|replace| (SELECTION CH#) |of| SEL |with| (ADD1 CH#))
(|replace| (SELECTION CHLIM) |of| SEL |with| (ADD1 ENDCH#))
(|replace| (SELECTION DCH) |of| SEL |with| (ADD1 (IDIFFERENCE ENDCH# CH#)))
(\TEDIT.DELETE SEL TEXOBJ) (* |Now| |delete| |the| |characters|)
)))

```

(TEEXEC.FILLBUFFER.LINEDELETE

(LAMBDA (TEXOBJ)

; Edited 30-May-91 19:18 by jds

(* |Delete| |last| |line| |entered| |by| |user| (^U, ^Q))

```

(DECLARE (USEDFREE RSNX RTBLSA \PRIMTERMSA TCLASS FILLTYPE \CURRENTRTD# \LINEBUF.OFD))
(PROG* (C (STARTINGEOF (TEXTPROP TEXOBJ 'STARTINGEOF))
  (SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
  (ENDCH# (SUB1 (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (COND
      ((ZEROP (|fetch| (SELECTION DCH) |of| SEL))
       (|fetch| (SELECTION CHLIM) |of| SEL))
      (T (SUB1 (|fetch| (SELECTION CHLIM) |of| SEL))))))
    (RIGHT (|fetch| (SELECTION CHLIM) |of| SEL))
    NIL)))
  (CH# ENDCH#))
  (COND
    ((ILEQ CH# (TEXTPROP (CAR (|fetch| (TEXTOBJ \WINDOW) |of| TEXOBJ))
      'STARTINGEOF))
     (FLASHWINDOW)
     (RETURN)))
    (|while| (AND (SETQ C (PROGN (\SETUPGETCH CH# TEXOBJ)
      (\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ)))
      (NEQ C (CHARCODE EOL))
      (IGREATERP CH# (TEXTPROP (CAR (|fetch| (TEXTOBJ \WINDOW) |of| TEXOBJ))
        'STARTINGEOF)))
      |do| (SETQ RSNX (\SYNCODE RTBLSA C))
      (|add| CH# -1)
      (COND
        ((EQ FILLTYPE READ.FT) (* |Invalidate| |paren| |count| |to| |force| |recalculation|)
         (|replace| (STREAM REVALIDATEFLG) |of| \LINEBUF.OFD |with| T)))
        (TEDIT.RESET.EXTEND.PENDING.DELETE SEL) (* |Make| |it| \a |normal| |selection| |again|.))
        (|replace| (SELECTION CH#) |of| SEL |with| (ADD1 CH#))
        (|replace| (SELECTION CHLIM) |of| SEL |with| (ADD1 ENDCH#))
        (|replace| (SELECTION DCH) |of| SEL |with| (ADD1 (IDIFFERENCE ENDCH# CH#)))
        (\TEDIT.DELETE SEL TEXOBJ) (* |Now| |delete| |the| |characters|)
      )))
  )))

```

(TEEXEC.PARENCOUNT

(LAMBDA (TEXOBJ)

; Edited 13-Jun-90 00:17 by mitani

```

(DECLARE (USEDFREE \LINEBUF.OFD))
(PROG ((STREAM (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))
  (|replace| (LINEBUFFER LPARCOUNT) |of| \LINEBUF.OFD |with| 0)
  (|replace| (LINEBUFFER LBRKCOUNT) |of| \LINEBUF.OFD |with| 0)
  (|replace| (LINEBUFFER LPARCOUNT) |of| \LINEBUF.OFD |with| 0)
  (|replace| (LINEBUFFER INSTRINGP) |of| \LINEBUF.OFD |with| NIL)
  (\SETUPGETCH (TEXTPROP TEXOBJ 'STARTINGEOF)
    TEXOBJ) (* |Don't| |include| |last| |character| |otherwise| |it| |will| |get|
    |counted| |twice|)
  (|for| I |from| (TEXTPROP TEXOBJ 'STARTINGEOF) |to| (SUB1 (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
    |do| (\INCPARENCOUNT (\SYNCODE RTBLSA (\BIN STREAM)))
  (|replace| (STREAM REVALIDATEFLG) |of| \LINEBUF.OFD |with| NIL)
  (* |Last| |thing| |is| |to| |reset| |the| |validation| |flag|, |and| |always| RETURN T |for| |AND's| |to| |work|)
  (RETURN T)))

```

(TEEXEC.PARENMATCH


```
(LAMBDA (TEXOBJ RSNX) ; Edited 13-Jun-90 00:17 by mitani
;; Returns CH# if matching left paren/bracket is found, else NIL
(DECLARE (USEDFREE \#CURRENTRDTBL# \TERM.OFD))
(PROG (MATCH? N CH CH1 CH2 (PCOUNT 0))
  (SELECTC RSNX
    (RIGHTPAREN.RC
      (SETQ CH (CAR (GETSYNTAX 'LEFTPAREN \#CURRENTRDTBL#)))
      (SETQ CH1 (CAR (GETSYNTAX 'RIGHTPAREN \#CURRENTRDTBL#))))
    (RIGHTBRACKET.RC
      (SETQ CH (CAR (GETSYNTAX 'LEFTBRACKET \#CURRENTRDTBL#)))
      (SETQ CH1 (CAR (GETSYNTAX 'RIGHTBRACKET \#CURRENTRDTBL#))))
    0)
  (|for| I |from| 0 |to| (IDIFFERENCE (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
    (TEXTPROP TEXOBJ 'STARTINGEOF))
  |do| (SETQ CH2 (TEEXEC.NTHBUFCHARBACK TEXOBJ I))
    (COND
      ((EQ CH2 CH1)
        (SETQ PCOUNT (ADD1 PCOUNT)))
      ((EQ CH2 CH)
        (SETQ PCOUNT (SUB1 PCOUNT))))
  |repeatuntil| (SETQ MATCH? (EQ PCOUNT 0)) |finally| (COND
    (MATCH?
      ;\CARET.DOWN \TERM.OFD
      (SETQ N (TEEXEC.NTHBACKCHNUM TEXOBJ I))))
  (RETURN N))))
```

(TEEXEC.FLASHCARET

(LAMBDA (TEXOBJ N) ; Edited 30-May-91 19:18 by jds

(* * |Flashes| |caret| |at| |char#| N, |used| |for| |flashing| |caret| |at| |matching| |paren/bracket| N N |is| NIL, |simply| |returns|)

```
(PROG (TSEL (WINDOW (|fetch| (TEXTOBJ \WINDOW) |of| TEXOBJ)))
  (COND
    ((NOT N)
      (RETURN)))
  (SETQ TSEL (|create| SELECTION |using| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
    CH# _ N CHLIM _ N DCH _ 0 POINT _ 'LEFT \TEXTOBJ _ TEXOBJ SET _ T)
  )
  (UNINTERRUPTABLY
    (|bind| (FIRSTTIME _ T) |for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXOBJ) |as| WIN |inside| WINDOW
    |as| L1 |inside| (|fetch| (SELECTION L1) |of| TSEL)
    |do| (COND
      (L1 (\EDIT.UPCARET CARET)
        (\FIXSEL TSEL TEXOBJ WIN)
        (COND
          (FIRSTTIME (SETQ FIRSTTIME NIL)
            (\CARET.FLASH? (|fetch| TCCARETDS |of| CARET)
              (|fetch| TCCARET |of| CARET)
              NIL NIL (|fetch| (SELECTION X0) |of| TSEL)
              (|fetch| (SELECTION Y0) |of| TSEL)))
            (T (\CARET.FLASH.AGAIN (|fetch| TCCARET |of| CARET)
              (|fetch| TCCARETDS |of| CARET)
              (|fetch| (SELECTION X0) |of| TSEL)
              (|fetch| (SELECTION Y0) |of| TSEL))))))
      (DISMISS 500)
      (|for| WIN |inside| WINDOW |as| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXOBJ)
      |do| (\EDIT.DOWNCARET CARET) (* |Display| |the| |caret| |at| |the| |typein| |point|))))))
```

(TEEXEC.TEXTSTREAM.TO.LINEBUF

(LAMBDA (TEXOBJ STARTINGEOF LINEBUF FILLTYPE) ; Edited 13-Jun-90 00:17 by mitani

;; Copy the contents of the current edit line into the line buffer, for transmission to the system.

```
(\SETUPGETCH (IMIN (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
  (ADD1 STARTINGEOF))
  TEXOBJ) ; Move to where the user's typed input starts in the edit buffer,
; and clear out the line buffer.
(SETFILEPTR LINEBUF 0)
(\SETEOFPTR LINEBUF 0)
(LET* ((TEXTSTREAM (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))
  (DRIBBLE (STREAMPROP TEXTSTREAM 'DRIBBLESTREAM)))
  (|while| (NOT (EOFP TEXTSTREAM))
    |do| (\LINEBUFBOUT LINEBUF (PROG ((C (\BIN TEXTSTREAM))
      (COND
        ((AND (NEQ FILLTYPE READC.FT)
          (EQ (CHARCODE ^V)
            C)) ; convert ^Vx to real CONTROL char
          (SETQ C (COND
            ((OR (AND (IGE Q (SETQ C (\BIN TEXTSTREAM))
              (CHARCODE A))
              (ILEQ C (CHARCODE Z))))
            (AND (IGE Q C (CHARCODE \a))
              (ILEQ C (CHARCODE \z))))
          ; CONVERT TO CONTROL CHAR
          (LOGAND C 31))
          (T C))))))
```

```
(COND
  (DRIBBLE ; Dribble output
    (\\OUTCHAR DRIBBLE C)))
(RETURN C))))
```

T))

(TEEXEC.FIX

(LAMBDA (STREAM LISPY TAIL) ; Edited 13-Jun-90 00:17 by mitani

(* |Inserts| |the| |text| |into| |the| |textstream| |and| |then| |positions| |to| |the| |end| |for| |user| |editing|)

```
(DECLARE (USEDFREE \#CURRENTRDTBL#))
(PROG ((TEXOBJ (TEXTOBJ STREAM))
  (RDTBL (COPYREADTABLE \#CURRENTRDTBL#))
  APPLYFLG
  (NCHARS 1))
  (SETSYNTAX (CHARCODE \")
    'OTHER RDTBL)
  (SETQ LISPY (REMOVE HISTSTRO LISPY)) (* |Remove| "<c.r.>")
  (COND
    (LISPY (TEXTPROP TEXOBJ 'STARTINGEOF (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
      (* |Set| |the| |new| STARTINGEOF |to| |the| |beginning| |of| |the|
        |text| |to| |insert|)
      (SETQ LISPY (CHCON (COND
        ((LISTP (CAR LISPY)) (* EVAL |format|)
          (CAR LISPY))
        (T (* APPLY |format|)
          (SETQ APPLYFLG LISPY)
          LISPY))
        T RDTBL))
      (COND
        (APPLYFLG (SETQ LISPY (REVERSE (CDR (REVERSE (CDR LISPY))))))
          (* |Remove| |surrounding| |parens| |brackets|)
        ))
      (|for| CH |inside| LISPY |do| (|add| NCHARS 1)
        (COND
          ((AND APPLYFLG (EQ CH (CHARCODE \ ))
            (OR (EQ (CAR (NTH LISPY NCHARS))
              (CHARCODE \ ( ))
              (EQ (CHARCODE [ ]
                (CAR (NTH LISPY NCHARS))))))
            (SETQ APPLYFLG NIL)
              (* |Remove| |space| |before| |a| |left| |paren| |bracket| |to| |put|
                |back| |in| APPLY |mode|)
            )
          (T (BKSYSBUF (CHARACTER CH))))))
        (* |Unread| |the| |characters| |back| |into| |the| |main| |keyboard|
          |buffer|)
        (STREAMPROP STREAM 'FIXFLG T) (* |Indicate| |to| TEEXEC.FILLBUFFER |we| |are| |in| |fix| |mode|)
        (RETURN))))))
```

(TEEXEC.NTHBUFCHARBACK

(LAMBDA (TEXOBJ N) ; Edited 13-Jun-90 00:17 by mitani

(* |Return| |the| |Nth| |character| |back| |from| |the| |current| |end| |of| |text.|
|Puts| |the| |pointer| |back| |at| |the| |end| |of| |the| |buffer|)

```
(PROG ((CHPOS (TEEXEC.NTHBACKCHNUM TEXOBJ N)) (* CHPOS |is| |the| |actual| |char| |position| |in| |the| |text.|)
  (COND
    ((OR (ILEQ CHPOS (TEXTPROP TEXOBJ 'STARTINGEOF))
      (ILEQ CHPOS 1)) (* |returns| NIL |if| |there| |is| |no| |char| |at| |that| |position|)
    (RETURN))
  (\\SETUPGETCH CHPOS TEXOBJ) (* |Setup| |position| |to| BIN |character|)
  (RETURN (\\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))))))
```

(TEEXEC.NTHBACKCHNUM

(LAMBDA (TEXOBJ N) ; Edited 30-May-91 19:18 by jds

(* |Converts| N |chars| |back| |from| |the| |end| |of| |the|
|selection| |to| |a| CH#)

```
(IDIFFERENCE (IMIN (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
  (SELECTQ (|fetch| (SELECTION POINT) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
    (LEFT (SUB1 (|fetch| (SELECTION CH#) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))))
    (RIGHT (|fetch| (SELECTION CHLIM) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)))
    NIL))
  N))
```

(TEEXEC.EOTP

(LAMBDA (TEXOBJ) ; Edited 30-May-91 19:18 by jds

(* |Test| |if| |the| |caret| |is| |at| |the| |end| |of| |the| |text|)

```
(PROG ((SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
  (RETURN (ILESSP (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
    (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
      (LEFT (|fetch| (SELECTION CH#) |of| SEL))
```

```
(RIGHT (|fetch| (SELECTION CHLIM) |of| SEL))
NIL))))))
```

(TEXEC.GETKEY

(LAMBDA (TEXOBJ)

; Edited 13-Jun-90 00:17 by mitani

;;; Return a character from the keyboard without echoing. If no key has been typed, update the screen if prior input-output didn't want to do it, flash the
;;; caret in all of the attached windows if the keyboard is attached to this process, ie, is the TTYDISPLAYSTREAM. Blocks until a key is typed with the
;;; keyboard attached to this process

```
(PROG ((SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
(|until| (\\WAITFORSYSBUFP 100) |do| (OR (|fetch| (TEXTOBJ EDITFINISHEDFLG) |of| TEXOBJ
(|while| (OR TEDIT.SELPENDING (|fetch| (TEXTOBJ EDITOPACTIVE)
|of| TEXOBJ)
|do| ; Don't anything while he's selecting or one of the lock-out ops is
; active.
(AND (OR (EQ TEDIT.SELPENDING TEXOBJ))
(TEDIT.FLASHCARET (|fetch| (TEXTOBJ CARET) |of| TEXOBJ)
)
(BLOCK))) ; Flash carets in all windows until a key is entered unless
; process is being terminated
(OR (|fetch| (TEXTOBJ EDITFINISHEDFLG) |of| TEXOBJ)
(COND
(|fetch| (TEXTOBJ TXTNEEDSUPDATE) |of| TEXOBJ)
; Screen needs updating
(\\SHOWSEL SEL NIL NIL)
(PRINTOUT PROMPTWINDOW "Needed updating." T)
(TEDIT.UPDATE.SCREEN TEXOBJ)
; Turn highlighting off of selection
(\\FIXSEL SEL TEXOBJ)
; Fix up the selection fields
(TEDIT.SCROLL? TEXOBJ)
; Scroll windows if necessary
(\\SHOWSEL SEL NIL T)
; Turn any selection back on
)))
(OR (|fetch| (TEXTOBJ EDITFINISHEDFLG) |of| TEXOBJ)
(PROGN (TEDIT.FLASHCARET (|fetch| (TEXTOBJ CARET) |of| TEXOBJ)
; Check for SHIFT/COPY/MOVE active
(TEXEC.CHSELPENDING TEXOBJ))))
(RETURN (AND (NOT (|fetch| (TEXTOBJ EDITFINISHEDFLG) |of| TEXOBJ)
(\\GETKEY))))))
```

(TEXEC.INSERTCHAR

(LAMBDA (TEXOBJ C)

; Edited 30-May-91 19:18 by jds

(* |Inserts| |a| |character| |into| |the| |textstream|. |If| |the| |character| |is| |being| |inserted| |prior| |to| |the| |current| |input|
|then| |the| |start| |of| |the| |current| |input| |pointer| |STARTINGEOF| |is| |incremented|.
|If| |echoing| |is| |off|, |then| |the| |character| |is| |inserted| |as| |"invisible"*)

```
(DECLARE (USEDFREE \\PRIMTERMTABLE \\TERM.OFD))
(PROG ((SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
(COND
((ILEQ (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
(LEFT (|fetch| (SELECTION CH#) |of| SEL))
(RIGHT (|fetch| (SELECTION CHLIM) |of| SEL))
NIL)
(TEXTPROP TEXOBJ 'STARTINGEOF))
(TEXTPROP TEXOBJ 'STARTINGEOF (ADD1 (TEXTPROP TEXOBJ 'STARTINGEOF))))
(* |If| |inserting| |text| |prior| |to| |current| |input| |move| |the| |start| |of| |input| |down| 1 |to| |compensate|)
)
(T (AND (IGREATERP (IMIN (|fetch| (SELECTION CH#) |of| SEL)
(|fetch| (SELECTION CHLIM) |of| SEL))
(TEXTPROP TEXOBJ 'STARTINGEOF))
(TEDIT.DO.BLUEPENDINGDELETE SEL TEXOBJ)))) (* |Only| |allow| |deletion| |of| |selected| |text| |if| |after| |start|
|of| |current| |input|)
(COND
(|fetch| ECHOFLG |of| \\PRIMTERMTABLE)
(TEDIT.\\INSERT (OR C (SETQ C (TEXEC.GETKEY TEXOBJ)))
SEL \\TERM.OFD))
(T (TEDIT.CARETLOOKS TEXOBJ '(INVISIBLE ON))
(TEDIT.\\INSERT (OR C (SETQ C (TEXEC.GETKEY TEXOBJ)))
SEL \\TERM.OFD)
(TEDIT.CARETLOOKS TEXOBJ '(INVISIBLE OFF))))
(RETURN C) (* |Return| |character| |inserted|)
)))
```

(TEXEC.DELETE

(LAMBDA (TEXOBJ START END)

; Edited 13-Jun-90 00:17 by mitani

(* |Deletes| |the| |chars| |in| |the| |textstream| |from| |START| |to|
|END|)

```
(LET ((TSEL (|create| SELECTION |using| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
      CH# _ START CHLIM _ (ADD1 END)
      POINT _ 'LEFT DCH _ (ADD1 (IDIFFERENCE END START))))
      (STREAM (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ)))
      (\\TEDIT.DELETE TSEL STREAM T))))
```

(TEEXEC.CHDEL1

(LAMBDA (TEXOBJ)

; Edited 30-May-91 19:19 by jds

(* |Delete| |the| |last| |character| |in| |the| |text| |stream,| |and| |return| |it,| |so| |we| |can| |decide| |what| |to| |do| |with| |it.))

(DECLARE (USEDFFREE \\LINEBUF.OFD))

```
(LET* ((SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)
      (TEXTLEN (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)
      (CH (COND
```

```
((SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
  (LEFT (IDIFFERENCE (|fetch| (SELECTION CH#) |of| SEL)
    1))
  (RIGHT (|fetch| (SELECTION CH#) |of| SEL)
  NIL))))))
```

(* CH = |character| |position| |at| |current| |location| |of| |caret|)

```
(COND
  ((NEQ CH (TEXTPROP TEXOBJ 'STARTINGEOF))
```

(* |don't| |allow| |deletion| |of| |text| |if| |at| |beginning| |of| |current| |input|)

(COND

```
((ILEQ (SELECTQ (|fetch| (SELECTION POINT) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))
  (LEFT (|fetch| (SELECTION CH#) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)))
  (RIGHT (SUB1 (|fetch| (SELECTION CHLIM) |of| (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)))
  NIL)
  (TEXTPROP TEXOBJ 'STARTINGEOF))
  (TEXTPROP TEXOBJ 'STARTINGEOF (SUB1 (TEXTPROP TEXOBJ 'STARTINGEOF))))
```

(* |If| |deleting| |text| |prior| |to| |current| |input| |move| |the| |start| |of| |input| |up| 1 |to| |compensate|)

))

(COND

((TEEXEC.EOTP TEXOBJ)

(T (|replace| (STREAM REVALIDATEFLG) |of| \\LINEBUF.OFD |with| T)

(* |Invalidate| |paren/bracket| |count| |since| |we| |are| |no| |longer| |at| |the| |end| |of| |the| |text|)

))

(PROG1 (PROGN (\\SETUPGETCH CH TEXOBJ)

(\\BIN (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ)))

(\\TEDIT.CHARDELETE TEXOBJ "" (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))))))

(TEEXEC.EQUAL

(LAMBDA (TEXOBJ)

; Edited 13-Jun-90 00:17 by mitani

(* |Test| |for| ? |or| ?= |and| |if| |so| RETURN T |else| RETURN NIL)

(DECLARE (USEDFFREE \\#CURRENTRDTBL#))

(PROG ((CH# 0)

CH FLG (TS (|fetch| (TEXTOBJ STREAMHINT) |of| TEXOBJ))

LST PTR FNAME TAIL TEMPFILE (STARTINGEOF (TEXTPROP TEXOBJ 'STARTINGEOF)))

(COND

```
((OR (AND (EQ (TEEXEC.NTHBUFCHARBACK TEXOBJ 0)
  (CHARCODE =))
  (EQ (TEEXEC.NTHBUFCHARBACK TEXOBJ 1)
  (CHARCODE ?))
  (FMEMB (TEEXEC.NTHBUFCHARBACK TEXOBJ 2)
  (CHCON ' | ' ( | ' [])))
  (AND (EQ (TEEXEC.NTHBUFCHARBACK TEXOBJ 0)
  (CHARCODE ?))
  (FMEMB (TEEXEC.NTHBUFCHARBACK TEXOBJ 1)
  (CHCON ' | ' ( | ' []))))
```

(RETURN T))

(T (RETURN))))))

(TEDIT.SCROLL?

(LAMBDA (TEXOBJ)

; Edited 30-May-91 19:19 by jds

(* |This| |function| |scrolls| |all| |of| |the| |windows| |if| |the| |caret| |is| |off-window| |in| |the| |selection| |window|)

(PROG ((SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ))

SELINE)

(|for| WIN |inside| (|fetch| (TEXTOBJ \\WINDOW) |of| TEXOBJ) |as| L1 |on| (|fetch| (SELECTION L1) |of| SEL)

|as| LN |on| (|fetch| (SELECTION LN) |of| SEL)

|do| (COND

```
((AND (EQ WIN (|fetch| (TEXTOBJ SELWINDOW) |of| TEXOBJ))
  (OR (NOT (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (CAR L1))
    (RIGHT (CAR LN))
    NIL))
  (ILEQ (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (|fetch| (LINEDESCRIPTOR YBOT) |of| (CAR L1))))
```

(LEFT (|fetch| (LINEDESCRIPTOR YBOT) |of| (CAR L1))))

```

(RIGHT (|fetch| (LINEDESCRIPTOR YBOT) |of| (CAR LN)))
0)
(|fetch| (REGION BOTTOM) |of| (DSPCLIPPINGREGION NIL WIN))))
(* |The| |caret| |is| |off-window| |in| |the| |selection| |window.| |Need| |to| |scroll| |it| |up| |so| |the| |caret| |is| |visible.|)
(|while| (OR (COND
  ((SETQ SELINE (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (CAR L1))
    (RIGHT (CAR LN))
    NIL))
  (ILESSP (|fetch| (LINEDESCRIPTOR YBOT) |of| SELINE)
    (|fetch| (TEXTOBJ WBOTTOM) |of| TEXOBJ)))
  (T (ILESSP (|fetch| (SELECTION Y0) |of| SEL)
    (|fetch| (TEXTOBJ WBOTTOM) |of| TEXOBJ))))
  (AND (IGEQ (|fetch| (SELECTION Y0) |of| SEL)
    (|fetch| (TEXTOBJ WTOP) |of| TEXOBJ))
    (NULL SELINE)))
|do| (* |The| |caret| |just| |went| |off-screen.|
  |Move| |it| |up| |some.|)
(|replace| (TEXTOBJ EDITOPACTIVE) |of| TEXOBJ |with| NIL)
(SCROLLW WIN 0 (LLSH (COND
  ((SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
    (LEFT (CAR L1))
    (RIGHT (CAR LN))
    NIL)
  (|fetch| (LINEDESCRIPTOR LHEIGHT)
    |of| (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
      (LEFT (CAR L1))
      (RIGHT (CAR LN))
      (SHOULDNT))))))
  (T 12))
1))))))

```

(TEEXEC.DISPLAYTEXT

(LAMBDA (TEXTOBJ CH FONT LINE XPOINT DS SEL)

; Edited 30-May-91 19:16 by jds

(* |This| |function| |does| |the| |actual| |displaying| |of| |typed-in| |text| |on| |the| |edit| |window.|)

```

(PROG ((TERMSA (|fetch| (TEXTOBJ TXTERMSA) |of| TEXTOBJ)
  DY)
  (DSPXPOSITION XPOINT DS)
  (COND
    (TERMSA

```

(* |Set| |the| |display| |stream| X |position|)

(* |Special| |terminal| |table| |for| |controlling| |character| |display.| |Use| |it.|)

```

(COND
  ((STRINGP CH)
    (|for| CHAR |instring| CH |do| (SELCHARQ CHAR
      (TAB (* |Put| |down| |white|)
        (BITBLT NIL 0 0 DS XPOINT (|fetch| (LINEDESCRIPTOR
          YBOT)
          |of| LINE)
          36
          (|fetch| (LINEDESCRIPTOR LHEIGHT)
            |of| LINE)
          'TEXTURE
          'REPLACE WHITESHADE)
          (RELMOVETO 36 0 DS))
        (CR (BITBLT NIL 0 0 DS XPOINT (|fetch| (LINEDESCRIPTOR
          YBOT)
          |of| LINE)
          (IMAX 6 (CHARWIDTH CHAR FONT))
          (|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
          'TEXTURE
          'REPLACE WHITESHADE))
          (\\DSPPRINTCHAR DS CHAR))))

```

```

(T (SELCHARQ CH
  (TAB (* |Put| |down| |white|)
    (BITBLT NIL 0 0 DS XPOINT (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
      36
      (|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
      'TEXTURE
      'REPLACE WHITESHADE)
      (RELMOVETO 36 0 DS))
    (CR (BITBLT NIL 0 0 DS XPOINT (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
      (IMAX 6 (CHARWIDTH CH FONT))
      (|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
      'TEXTURE
      'REPLACE WHITESHADE))
      (\\DSPPRINTCHAR DS CH))))

```

(* N'o |special| |handling;| |just| |use| |native| |character| |codes|)

```

(COND
  ((STRINGP CH)
    (|for| CHAR |instring| CH |do| (SELCHARQ CHAR
      (TAB (* |Put| |down| |white|)
        (BITBLT NIL 0 0 DS (DSPXPOSITION NIL DS)
          (|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)

```

```

36
(|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
' TEXTURE
'REPLACE WHITESHAE)
(RELMOVETO 36 0 DS))
(CR (BITBLT NIL 0 0 DS (DSPXPOSITION NIL DS)
(|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
(IMAX 6 (CHARWIDTH CHAR FONT))
(|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
' TEXTURE
'REPLACE WHITESHAE))
(BLTCHAR CHAR DS)))
(T (SELCHARQ CH
(TAB
(BITBLT NIL 0 0 DS (DSPXPOSITION NIL DS)
(|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
36
(|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
' TEXTURE
'REPLACE WHITESHAE)
(RELMOVETO 36 0 DS))
(CR
(BITBLT NIL 0 0 DS (DSPXPOSITION NIL DS)
(|fetch| (LINEDESCRIPTOR YBOT) |of| LINE)
(IMAX 6 (CHARWIDTH CH FONT))
(|fetch| (LINEDESCRIPTOR LHEIGHT) |of| LINE)
' TEXTURE
'REPLACE WHITESHAE))
(BLTCHAR CH DS))))))
(* |Put| |down| |white|)
(* |Blank| |out| |the| |CR's| |width|.))

```

(\\TEEXEC.TEXTBOUT

(LAMBDA (STREAM BYTE)

; Edited 30-May-91 19:19 by jds

(* * D'o BOUT |to| |a| |text| |stream,| |which| |is| |an| |insertion| |at| |the| |end| |of| |text| |pointer|)

(UNINTERRUPTABLY

```

(PROG ((TEXOBJ (|fetch| (TEXTSTREAM TEXTOBJ) |of| STREAM))
CH# WINDOW TEXTLEN SEL STARTINGEOF)
(SETQ STARTINGEOF (ADD1 (TEXTPROP TEXOBJ 'STARTINGEOF)))

```

(* |Insertion| |point| |for| |output| -
|is| |actually| |the| |beginning| |of| |the| |next/current| |text| |being| |entered| |to| |return| |to| |READ|)

```

(TEXTPROP TEXOBJ 'STARTINGEOF STARTINGEOF) (* |This| |adds| 1 |to| |the| |previous| |value| |on| |the| |property|
|list|)
(SETQ TEXTLEN (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)) (* |The| |length| |of| |the| |total| |text| |in| |chars|)
(SETQ WINDOW (|fetch| (TEXTOBJ \\WINDOW) |of| TEXOBJ))
(SETQ SEL (|fetch| (TEXTOBJ SEL) |of| TEXOBJ)) (* |The| |current| |selection|)
(* COND ((NOT (LDIFFERENCE
(|fetch| (SELECTION L1) |of| SEL)
(QUOTE (NIL)))) (RETURN)))
(* |Return| |if| |caret| |out| |of| |bounds,| |ie,| |user| |scrolls| |past|

```

|end| |of| |text|)

```

(SETQ CH# (OR STARTINGEOF (|fetch| (SELECTION CH#) |of| SEL)))
(AND WINDOW (|for| L1 |in| (|fetch| (SELECTION L1) |of| SEL) |as| LN |in| (|fetch| (SELECTION LN) |of| SEL)
|do| (* |Mark| |changed| |lines| |as| |DIRTY.))
(COND
(L1 (|replace| (LINEDESCRIPTOR DIRTY) |of| L1 |with| T))
(LN (|replace| (LINEDESCRIPTOR DIRTY) |of| LN |with| T))))
(\\INSERTCH BYTE CH# TEXOBJ)
(|replace| (TEXTOBJ TXTNEEDSUPDATE) |of| TEXOBJ |with| T)
(AND WINDOW (\\TEEXEC.TEXTBOUT1 TEXOBJ STREAM BYTE CH# SEL))))

```

(\\TEEXEC.TEXTBOUT1

(LAMBDA (TEXOBJ STREAM BYTE CH# SEL)

; Edited 30-May-91 19:19 by jds

```

(PROG ((THISLINE (|fetch| (TEXTOBJ THISLINE) |of| TEXOBJ))
PS PC OFFST)

```

::: Update stream CHARPOSITION for calls to POSITION ie, GAINSPACE

```

(|add| (|fetch| (SELECTION CH#) |of| SEL)
1) ; These must be here, since SELs are valid even without a
; window.
(|replace| (SELECTION CHLIM) |of| SEL |with| (|fetch| (SELECTION CH#) |of| SEL))
(|replace| (SELECTION POINT) |of| SEL |with| 'LEFT)
(|replace| (SELECTION DCH) |of| SEL |with| 0)
(|replace| (SELECTION SELKIND) |of| SEL |with| 'CHAR)
(|for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXOBJ) |do| (\\EDIT.UPCARET CARET))
(\\TEEXEC.TEXTBOUT2 TEXOBJ STREAM BYTE CH# SEL)
(|for| CARET |inside| (|fetch| (TEXTOBJ CARET) |of| TEXOBJ) |do| ;; Allow carets to flash again.
(|replace| (TEDITCARET TCFORCEUP) |of| CARET
|with| NIL))
(|replace| (SELECTION ONFLG) |of| SEL |with| T)
(|replace| (DESC |of| THISLINE) |with| NIL)

```

:: SO that this line of text is run thru the formatter again before anything interesting that depends on it being right (like scrolling the window)

::: Fix up the TEXTSTREAM so that the FILEPTR looks like it ought to after the BOUT, even though we've been updating the screen (which usually moves the fileptr....)

```
(SETQ PS (|fetch| (PIECE PSTR) |of| (SETQ PC (|fetch| (TEXTOBJ \\INSERTPC) |of| TEXOBJ)))
; This piece resides in a STRING. Because it's newly 'typed'
; material.
(|replace| (TEXTSTREAM PIECE) |of| STREAM |with| PC) ; Remember the current piece for others.
(|replace| (TEXTSTREAM PCNO) |of| STREAM |with| (|fetch| (TEXTOBJ \\INSERTPCNO) |of| TEXOBJ)) ; And which number piece this is.
(|replace| (STREAM CPPTR) |of| STREAM |with| (ADDBASE (|fetch| (STRINGP BASE) |of| PS)
(LRSH (SETQ OFFST (|fetch| (STRINGP OFFST) |of| PS))
1)))
; Pointer to the actual characters in the string (allowing for
; substrings.)
(|replace| (STREAM CPAGE) |of| STREAM |with| 0)
(|replace| (STREAM COFFSET) |of| STREAM |with| (IPLUS (|replace| (TEXTSTREAM PCSTARTCH) |of| STREAM
|with| (LOGAND 1 OFFST))
(|fetch| (TEXTOBJ \\INSERTLEN) |of| TEXOBJ)))
(|replace| (TEXTSTREAM PCSTARTPG) |of| STREAM |with| 0) ; Page # within the 'file' where this piece starts
(|replace| (STREAM CBUFSIZE) |of| STREAM |with| (|fetch| (STREAM COFFSET) |of| STREAM))
(|replace| (STREAM EPAGE) |of| STREAM |with| 1)
(|replace| (TEXTSTREAM CHARSLEFT) |of| STREAM |with| 0) ; We're, perforce, at the end of the piece.
(|replace| (TEXTSTREAM REALFILE) |of| STREAM |with| NIL) ; We're not on a file....

)))
```

(\\TEEXEC.TEXTBOUT2

```
(LAMBDA (TEXOBJ STREAM BYTE CH# SEL) ; Edited 30-May-91 19:19 by jds
(PROG (YFLG)
(|for| WIN |in| (|fetch| (TEXTOBJ \\WINDOW) |of| TEXOBJ) |as| L1 |on| (|fetch| (SELECTION L1) |of| SEL)
|as| LN |on| (|fetch| (SELECTION LN) |of| SEL) |do| (COND
((CAR L1)
(|add| (|fetch| (LINEDESCRIPTOR CHARLIM)
|of| (CAR L1))
1)
(|add| (|fetch| (LINEDESCRIPTOR CHARTOP)
|of| (CAR L1))
1)
(COND
(EQ WIN (|fetch| (TEXTOBJ SELWINDOW)
|of| TEXOBJ))
(SETQ YFLG (ILESSP (|fetch| (LINEDESCRIPTOR
YBASE)
|of| (CAR L1))
0))))))
(COND
((OR (IGREATERP (PLUS (|fetch| (SELECTION X0) |of| SEL)
(Charwidth byte (|fetch| CLFONT |of| (|fetch| (TEXTOBJ CARETLOOKS) |of| TEXOBJ))))
(IDIFFERENCE (|fetch| (TEXTOBJ WRIGHT) |of| TEXOBJ)
16))
(IEQP BYTE (CHARCODE EOL))
(IEQP BYTE (CHARCODE CR))
(ILESSP CH# (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
YFLG))
(* |gone| |off| |the| |edge| |of| |the| |line| |Or| |not| |at| |end| |of| |text| |reformat| |and| |add| |new| |line|)
(\\TEEXEC.TEXTBOUT3 TEXOBJ STREAM BYTE CH# SEL))
(T (* |Display| |text| |on| |same| |line| |without| |updating| |entire|
|screen|)
(\\TEEXEC.TEXTBOUT4 TEXOBJ STREAM BYTE CH# SEL))))))
```

(\\TEEXEC.TEXTBOUT3

```
(LAMBDA (TEXOBJ STREAM BYTE CH# SEL) ; Edited 13-Jun-90 00:17 by mitani
(* |Updates| |the| |screen| |if| |necessary| |and| |checks| |for| |exceeding| |bufferlimit| |size|)
(PROG (OCHLIM)
(COND
((IGEQ CH# (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ))
(\\SHOWSEL SEL NIL NIL)
(TEDIT.UPDATE.SCREEN TEXOBJ) (* |Only| |update| |screen| |if| |at| |end| |of| |text|)
(\\FIXSEL SEL TEXOBJ NIL)
(TEDIT.SCROLL? TEXOBJ)
(\\SHOWSEL SEL NIL T)
(COND
((AND (NEQ TEEXEC.BUFFERLIMIT 0)
(IGREATERP (SETQ OCHLIM (IDIFFERENCE (GETEOPTR STREAM)
TEEXEC.BUFFERLIMIT))
0))
(|replace| (TEXTOBJ \\INSERTPCVALID) |of| TEXOBJ |with| NIL)
(TEEXEC.DELETE TEXOBJ 1 (IMAX OCHLIM (LRSH TEEXEC.BUFFERLIMIT 3))))))
```

(* |Remove| |the| |top| |1/8| |or| |at| |least| |the| |number| |of| |chars| |exceeding| TEXEC.BUFFERLIMIT.
|f| TEXEC.BUFFERLIMIT |is| 0\\, |allow| |infinite| |size|)

((\\FIXSEL SEL TEXOBJ))))))

(\\TEXEC.TEXTBOUT4

(LAMBDA (TEXOBJ STREAM BYTE CH# SEL)

; Edited 30-May-91 19:19 by jds

(* * |Display| |character| |on| |same| |line| |with| |no| |reformatting| |screen|)

(COND

((IGEQ CH# (|fetch| (TEXTOBJ TEXTLEN) |of| TEXOBJ)

(* |Don't| |display| |character| |unless| |the| |output| |is| |at| |the| |end.| \\TEXEC.GETKEY |will| |update| |the| |screen| |when|
|the| |next| |input| |is| |called| |for.|)

(|for| WIN |in| (|fetch| (TEXTOBJ \\WINDOW) |of| TEXOBJ) |as| L1 |on| (|fetch| (SELECTION L1) |of| SEL) |as| LN
|on| (|fetch| (SELECTION LN) |of| SEL) |do| (COND

((AND (CAR L1)
(IGEQ (|fetch| (LINEDESCRIPTOR YBASE)
|of| (CAR L1))
0))
(TEXEC.DISPLAYTEXT TEXOBJ BYTE (|fetch| CLFONT
|of| (|fetch| (TEXTOBJ
CARETLOOKS
)
|of| TEXOBJ))
(CAR L1)
(DSPXPOSITION NIL WIN)
(WINDOWPROP WIN 'DSP)
SEL) (* |Print| |out| |the| |character| |on| |the| |screen|)
(|replace| (SELECTION X0) |of| SEL |with| (DSPXPOSITION NIL WIN)
)
(|replace| (SELECTION XLIM) |of| SEL |with| (DSPXPOSITION NIL
WIN))))))

(\\TEXEC.SELFN

(LAMBDA (TEXOBJ SEL SELMODE CONTROL)

; Edited 30-May-91 19:19 by jds

(* * |This| |function| |gets| |called| |via| |the| SELFN |property| |on| |the| TEXTOBJ, |and| |restricts| |the| |selection| |to| |the|
|current| |input|)

(PROG NIL
(COND

(* |Make| |sure| |there| |is| |really| \\a |selection|)

((AND (FMEMB SELMODE ' (NORMAL PENDINGDEL MOVE DELETE))
(|fetch| (SELECTION SET) |of| SEL)
(ILESSP (SELECTQ (|fetch| (SELECTION POINT) |of| SEL)
(LEFT (|fetch| (SELECTION CH#) |of| SEL))
(RIGHT (|fetch| (SELECTION CHLIM) |of| SEL))
NIL)
(ADD1 (TEXTPROP TEXOBJ ' STARTINGEOF))))

(RETURN 'DON'T) (* |Don't| |allow| |selection| |if| |selecting| |text| |prior| |to| |current|
|input|)
)
(T (RETURN T)) (* |else| |return| T)

)))

(TEXEC.PRINTARGS

(LAMBDA (FN ARGS ACTUALS ARGTYPE)

; Edited 3-Mar-87 15:06 by raf

(PROG (TYPE REMARGS)
(\\CARET.DOWN)

:: Prints args to fn, matching up with ACTUALS, if supplied. Do this in a way that lets us keep track of where we are.

(TTPRIN1 " "
(TTPRIN2 FN)
(COND
((NOT ARGS))
(T (COND
((COND
((EQ ACTUALS T)
(SETQ ACTUALS (TTYIN.READ?=ARGS))
(T ACTUALS))
(COND
((CDR ACTUALS)
(TTCRLF))
(T (TTPRIN1 " ")))
(|bind| MODE |while| ACTUALS
|do| (COND
((EQ MODE '&KEY)
(TTPRIN1COMMENT " = ")
(TTPRIN2 (|pop| ACTUALS))
(TTPRIN1 " "
(TTPRIN2 (|pop| ACTUALS))
(TTCRLF)


```

(GO $$ITERATE))
(NULL ARGS)
(TTPRIN1COMMENT "+ ")
(NLISTP ARGS)
(TTPRIN1COMMENT (CONCAT " . " ARGS " = "))
(|while| ACTUALS |do| (TTPRIN2 (|pop| ACTUALS))
(TTPRIN1 " "))
(TTCRLF)
(SETQ ARGS)
(RETURN)
(T (SELECTQ (CAR ARGS)
(&REST &BODY)
(TTPRIN1COMMENT (CL:FORMAT NIL " ~A ~A =" (|pop| ARGS)
(|pop| ARGS)))
(|while| ACTUALS |do| (TTPRIN2 (|pop| ACTUALS))
(TTPRIN1 " "))
(TTCRLF)
(RETURN))
(&ALLOW-OTHER-KEYS
(TTPRIN1COMMENT (|pop| ARGS))
(TTCRLF)
(GO $$ITERATE))
(&OPTIONAL (TTPRIN1COMMENT (CL:FORMAT NIL "~A ~A" (SETQ MODE (|pop| ARGS))
(|pop| ARGS))))
(&KEY (SETQ MODE (CAR ARGS))
(|while| (AND ARGS (NOT (FMEMB (CAR ARGS)
CL:LAMBDA-LIST-KEYWORDS)))
|do| (TTPRIN1 " ")
(TTPRIN1COMMENT (|pop| ARGS)))
(TTCRLF)
(GO $$ITERATE))
(TTPRIN1COMMENT (|pop| ARGS))))))
(TTPRIN1COMMENT " = ")
(TTPRIN2 (CAR ACTUALS)
2 4)
(SETQ ACTUALS (CDR ACTUALS))
(TTCRLF)))
(|while| ARGS |do| (|if| (NLISTP ARGS)
|then| (TTPRIN1COMMENT (CONCAT " . " ARGS))
(TTCRLF)
(SETQ ARGS)
(RETURN))
(TTPRIN1 " ")
(TTPRIN1COMMENT (|pop| ARGS))))))
(TTPRIN1 " ")
(COND
((SETQ TYPE (SELECTQ (OR ARGTYPE (ARGTYPE FN))
(1 'NL)
(3 'NL*)
NIL))) ; indicate arg type
(TTPRIN1COMMENT (CONCAT " {" TYPE "}"))))))

```

(TEEXEC.PROCENTRYFN

```

(LAMBDA (NEWPROCESS OLDPROCESS) ; Edited 12-Dec-88 15:18 by jds
;; TEEXEC'S PROCESS.ENTRYFN which disarms the DELELTE key interrupt so it can use it as a linedelete key
(COND
((NOT (PROCESSPROP NEWPROCESS 'INTERRUPTS))
(PROCESSPROP NEWPROCESS 'INTERRUPTS (RESET.INTERRUPTS (REMOVE '(127 RUBOUT T)
(LISPINTERRUPTS)
T))))))

```

(TEEXEC.PROCEXITFN

```

(LAMBDA (THISP NEWP) (* AJB " 6-Feb-86 15:25")
(* |Re-arm| DELETE |key| |to| |whatever| |it| |was| |before|)
(RESET.INTERRUPTS (PROCESSPROP THISP 'INTERRUPTS))
(PROCESSPROP THISP 'INTERRUPTS NIL))
)

```

;; Code to support a TEEXEC lisp 'listener'

(DEFINEQ

(TEEXEC

```

(LAMBDA (REGION PROMPT MENUFN) ; Edited 8-Feb-89 16:55 by jds
;; Create an Interlisp executive window with TEdit as the editor behind it.
(PROG (HANDLE WINDOW TEDITSTREAM LINEBUFFER)
(SETQ WINDOW (CREATEW REGION "TEEXEC (Version 20.2)")) ; Create a window for the TEdit-based listener
(SETQ TEDITSTREAM (TEEXEC.OPENTEXTSTREAM WINDOW (OR MENUFN 'TEEXEC.DEFAULT.MENUFN))) ; Create a TEdit stream for the TEEXEC and fill in our
; non-standard fields
(|replace| (STREAM LINELENGTH) |of| TEDITSTREAM |with| (|fetch| (STREAM LINELENGTH) |of| (|fetch| (WINDOW DSP)
|of| WINDOW))))

```



```

\ " @ @ @ A O O O O H @ @ @ @ @ @ @ \ "
\ " @ @ @ C O O O O H @ @ @ @ @ @ @ \ "
\ " @ @ @ C O O O O H C @ @ @ @ @ @ @ \ "
\ " @ @ @ C O O O O L B @ @ @ @ @ @ @ \ "
\ " @ @ @ A O O O O N F @ @ @ @ @ @ @ \ "
\ " @ @ @ A O O O O N L @ @ @ @ @ @ @ \ "
\ " @ @ @ A O O O O O L @ @ @ @ @ @ @ \ "
\ " @ @ @ A O O O O O H @ @ @ @ @ @ @ \ "
\ " O O O O O O O O O O O O H @ @ @ @ @ @ @ \ "
\ " O O O O O O O O O O O O @ @ @ @ @ @ @ \ "
\ " O O O O O O O O O O O O N @ @ @ @ @ @ @ \ "
\ " O O O O O O O O O O O O O L @ @ @ @ @ @ @ \ "
\ " O O O O O O O O O O O O O H \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " O O O O O O O O O O O O O O \ "
\ " L O O O O O O O O O O O O O O \ "
\ " L A O O O O O O O O O O O O O O \ "
\ " @ @ @ C O O O O O O O O O O O O O \ "
\ " @ @ @ G H @ @ @ @ @ @ @ @ @ @ @ C \ "
\ " @ @ @ C H @ @ @ @ @ @ @ @ @ @ @ C \ "
\ " @ @ @ C H @ @ @ @ @ @ @ @ @ @ @ C \ "
\ " @ @ @ A H @ @ @ @ @ @ @ @ @ @ @ C \ " }
      ( 8 30 60 60 ) )
    " )

```

(RPAQ **TEXEC.ICON.TITLE.REGION** (CREATE REGION BOTTOM _ 55 LEFT _ 6 WIDTH _ 77 HEIGHT _ 16))

(RPAQ **TEXEC.ICON.FONT** (FONTCREATE 'HELVETICA 8 'BOLD))

```

(RPAQ TEXEC.DEFAULT.MENU
  (TEXEC.CREATEMENU ' ( ( |Put| ' |Put| NIL (SUBITEMS |Plain-Text| |Old-Format|) )
    ( |Include| ' |Include| )
    ("Forward Find" ' |ForwardFind| )
    ("Backward Find" ' |BackwardFind| )
    ( |Limit| ' |Limit| ) ) ) )

```

(RPAQQ **|BackgroundMenu|** NIL)

(RPAQ? **TEXEC.BUFFERLIMIT** 10000)

FUNCTION INDEX

TEDIT.SCROLL?	20	TEXEC.FILLBUFFER.WORDDELETE	15	TEXEC.PRINTARGS	24
TEXEC	25	TEXEC.FIND.BACKWARD	8	TEXEC.PROCENTRYFN	25
TEXEC.?EQUAL	20	TEXEC.FIND.FORWARD	7	TEXEC.PROCEXITFN	25
TEXEC.BACKSKREAD	1	TEXEC.FIX	18	TEXEC.SHRINK.ICONCREATE	10
TEXEC.CHSELPEINDING	14	TEXEC.FLASHCARET	17	TEXEC.TEXTSTREAM.TO.LINEBUF	17
TEXEC.CREATEMENU	4	TEXEC.GET	4	TEXEC.\\CHDEL1	20
TEXEC.DEFAULT.MENUFN	2	TEXEC.GETKEY	19	TTEXEC	26
TEXEC.DELETE	19	TEXEC.INCLUDE	5	\\TEXEC.SELFN	24
TEXEC.DISPLAYTEXT	21	TEXEC.INSERTCHAR	19	\\TEXEC.TEXTBOUT	22
TEXEC.DO?CMD	3	TEXEC.MENU.WHENHELDFN	10	\\TEXEC.TEXTBOUT1	22
TEXEC.EOTP	18	TEXEC.NTHBACKCHNUM	18	\\TEXEC.TEXTBOUT2	23
TEXEC.FILLBUFFER	10	TEXEC.NTHBUFCHARBACK	18	\\TEXEC.TEXTBOUT3	23
TEXEC.FILLBUFFER.CHARDELETE	15	TEXEC.OPENTEXTSTREAM	2	\\TEXEC.TEXTBOUT4	24
TEXEC.FILLBUFFER.LINEDELETE	16	TEXEC.PARENCOUNT	16		
TEXEC.FILLBUFFER.TCLASS	13	TEXEC.PARENMATCH	16		

VARIABLE INDEX

BackgroundMenu 	30	TEXEC.DEFAULT.MENU	30	TEXEC.ICON.TITLE.REGION	30
TEXEC.BUFFERLIMIT	30	TEXEC.ICON.FONT	30		
