File created: 21-Mar-2024 10:49:49 **{DSK}<Users>kaplan>Local>medley3.5>working-medley>library>tedit>TEDIT-SELECTION.;461**

edit by: rmk

changes to: (FNS TEDIT.SCANSEL TEDIT.XYTOCH TEDIT.SELECT)
(VARS TEDIT-SELECTIONCOMS)

previous date: 20-Mar-2024 11:08:55 {DSK}<Users>kaplan>Local>medley3.5>working-medley>library>tedit>TEDIT-SELECTION.;453

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-SELECTIONCOMS
       ((DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS SELECTION SELPIECES)
                                                 (CONSTANTS (COPYSELSHADE 30583)
                                                        (COPYLOOKSSELSHADE 30583)
                                                        (EDITMOVESHADE -1)
                                                        (EDITGRAY 32800))
                                                 (MACROS WITHINLINEP LINESELECTEDP)
                                                 (MACROS GETSEL SETSEL FGETSEL FSETSEL)
                                                 (GLOBALVARS TEDIT.EXTEND.PENDING.DELETE)
                                                 (GLOBALVARS TEDIT.SELECTION TEDIT.SHIFTEDSELECTION
                                                        TEDIT.MOVESELECTION TEDIT.COPYLOOKSSELECTION
                                                        TEDIT.DELETESELECTION)
                                                 (I.S.OPRS inselpieces)))
        (INITRECORDS SELECTION SELPIECES)
        (FNS \TEDIT.SELECTION.DEFPRINT)
        (FNS \TEDIT.SET.GLOBAL.SELECTIONS)
        (P (\TEDIT.SET.GLOBAL.SELECTIONS))
        (FNS \TEDIT.SELECTED.PIECES \TEDIT.FIND.PROTECTED.END \TEDIT.FIND.PROTECTED.START \TEDIT.WORD.BOUND)
        (INITVARS (TEDIT.EXTEND.PENDING.DELETE T))
                                                                ; Setting for a "Laurel" mode
        (COMS                                                   ; Selection manipulating code
              (FNS \TEDIT.EXTEND.SEL \TEDIT.SELECT \TEDIT.SCAN.LINE \TEDIT.SCAN.LINE.WORD
                   \TEDIT.SELECT.LINE.SCANNER \TEDIT.SELECT.OBJECT)
              (FNS \TEDIT.FIXSEL \TEDIT.CHTOX \TEDIT.COLLECTSELS \TEDIT.SELECTION.UNSET)
              (FNS \TEDIT.RESET.EXTEND.PENDING.DELETE \TEDIT.SET.SEL.LOOKS)
              (FNS \TEDIT.SHOWSEL \TEDIT.SHOWSEL.HILIGHT \TEDIT.UPDATE.SHOWSEL \TEDIT.REFRESH.SHOWSEL
                   \TEDIT.UPDATE.SEL \TEDIT.SEL.L1 \TEDIT.SEL.LN \TEDIT.SEL.DELETEDCHARS)
              (FNS \TEDIT.COPYSEL \TEDIT.SEL.CHANGED?))
        ;; SELPIECES
        (FNS \TEDIT.SELPIECES \TEDIT.SELPIECES.COPY \TEDIT.SELPIECES.CONCAT \TEDIT.SELPIECES.CHARTRANSFORM
             \TEDIT.SELPIECES.FROM.STRING \TEDIT.SELPIECES.TO.STRING)
        ;; User entries to the selection code
        (FNS TEDIT.XYTOCH TEDIT.GETPOINT TEDIT.GETSEL TEDIT.GETSEL.PARA TEDIT.MAKESEL TEDIT.SCANSEL
             TEDIT.SET.SEL.LOOKS TEDIT.SETSEL TEDIT.SHOWSEL TEDIT.SEL.AS.STRING TEDIT.SEL.AS.SEXPR
             TEDIT.SELECTALL)))

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(DATATYPE SELECTION (;; Description of a piece of selected text for TEdit.  Text has to be selected before it can be operated on by the user. The
                     ;; caret is to the left of CH# if POINT is LEFT, to the keft of CHLIM if POINT is RIGHT.

                     ;; If DCH > 0, highlighting goes from CH# to (SUB1 CHLIM = (SUB1 (IPLUS CH# DCH)).

                     ;; If DCH=0, this is a caret-only selection, with no highlighting.  In that case CHLIM=(ADD1 CH#) and POINT essentially
                     ;; indicates whether the caret blinks before or after CH#.
                     NIL                                        ; Was Y0: Y value of topmost line of selection
                     X0                                         ; X value of left edge of selection on the first line
                     NIL                                        ; Was DX: Width of the selection, if it's on one line.
                     CH#                                        ; CH# of the first selected character
                     XLIM                                       ; X value of right edge of last selected character on the last line
                     CHLIM                                      ; Last character is at (SUB1 CHLIM)
                     DCH                                        ; # of characters selected (can be zero, for empty/point
                                                                ; selection.) This controls highlighting
                     L1                                         ; -> line descriptor for the line where the first selected character
                                                                ; is
                     LN                                         ; -> line descriptor for the line which contains the end of the
                                                                ; selection
                     NIL                                        ; Was YLIM: Y value of the bottom of the line that ends the
                                                                ; selection
                     POINT                                      ; Which end should the caret appear at?  (LEFT or RIGHT)
                     (SET FLAG)                                 ; T if this selection is real;  NIL if not
                     (SELTEXTOBJ FULLXPOINTER)                  ; TEXTOBJ that describes the selected text
                     SELKIND                                    ; What kind of selection?  CHAR or WORD or LINE or PARA
                     HOW                                        ; SHADE used to highlight this selection
```

```
                    HOWHEIGHT                                   ; Height of the highlight (1 usually, full line for delete selection...)
                    (HASCARET FLAG)                             ; T if there should be a caret for this selection
                    SELOBJ                                      ; If this selection is inside an object, which object?
                    (ONFLG FLAG)                                ; T if the selection is highlighted on the screen, else NIL
                    SELOBJINFO                                  ; A Place for the selected object to put info about selection inside
                                                                ; itself.

                    )
      (INIT (DEFPRINT 'SELECTION (FUNCTION \TEDIT.SELECTION.DEFPRINT)))
      [ACCESSFNS (DX (AND (FIXP (fetch (SELECTION X0) of DATUM))
                          (FIXP (fetch (SELECTION XLIM) of DATUM))
                          (IDIFFERENCE (fetch (SELECTION XLIM) of DATUM)
                                 (fetch (SELECTION X0) of DATUM]
      SET _ NIL HOW _ BLACKSHADE HOWHEIGHT _ 1 HASCARET _ T X0 _ 0 POINT _ 'LEFT L1 _ (LIST NIL)
      LN _ (LIST NIL))

(DATATYPE SELPIECES (SPFIRST SPLAST SPLEN SPFIRSTCHAR SPLASTCHAR))
)

(/DECLAREDATATYPE 'SELECTION
      '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG
               FULLXPOINTER POINTER POINTER POINTER FLAG POINTER FLAG POINTER)
      ;; ---field descriptor list elided by lister---
      '34)

(DEFPRINT 'SELECTION (FUNCTION \TEDIT.SELECTION.DEFPRINT))

(/DECLAREDATATYPE 'SELPIECES '(POINTER POINTER POINTER POINTER POINTER)
      ;; ---field descriptor list elided by lister---
      '10)

(DECLARE%: EVAL@COMPILE

(RPAQQ COPYSELSHADE 30583)

(RPAQQ COPYLOOKSSELSHADE 30583)

(RPAQQ EDITMOVESHADE -1)

(RPAQQ EDITGRAY 32800)

(CONSTANTS (COPYSELSHADE 30583)
      (COPYLOOKSSELSHADE 30583)
      (EDITMOVESHADE -1)
      (EDITGRAY 32800))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS WITHINLINEP MACRO (OPENLAMBDA (CHNO LINE)
                              (AND (IGEQ CHNO (fetch (LINEDESCRIPTOR LCHAR1) of LINE))
                                   (ILEQ CHNO (fetch (LINEDESCRIPTOR LCHARLIM) of LINE))
                                   LINE)))

(PUTPROPS LINESELECTEDP MACRO [OPENLAMBDA (L CH# CHLIM)
                                (AND (IGEQ CHLIM (GETLD L LCHAR1))
                                     (ILEQ CH# (FGETLD L LCHARLIM])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETSEL MACRO ((S FIELD)
                        (fetch (SELECTION FIELD) of S)))

(PUTPROPS SETSEL MACRO ((S FIELD NEWVALUE)
                        (replace (SELECTION FIELD) of S with NEWVALUE)))

(PUTPROPS FGETSEL MACRO ((S FIELD)
                         (ffetch (SELECTION FIELD) of S)))

(PUTPROPS FSETSEL MACRO ((S FIELD NEWVALUE)
                         (freplace (SELECTION FIELD) of S with NEWVALUE)))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.EXTEND.PENDING.DELETE)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.SELECTION TEDIT.SHIFTEDSELECTION TEDIT.MOVESELECTION TEDIT.COPYLOOKSSELECTION
      TEDIT.DELETESELECTION)
)
```

```
(DECLARE%: EVAL@COMPILE

(I.S.OPR 'inselpieces NIL '[SUBST (GETDUMMYVAR)
                                 '$$SELPIECES
                                 '(bind $$SPFIRST $$SPLAST $$SPLENGTH $$SELPIECES _ BODY
                                     declare (LOCALVARS $$SELPIECES $$SPFIRST $$SPLAST $$SPLENGTH)
                                     first [SETQ I.V. (SETQ $$SPFIRST (\DTEST (OR (fetch (SELPIECES SPFIRST)
                                                                                    of $$SELPIECES)
                                                                                  (GO $$OUT))
                                                                              'PIECE]
                                         (SETQ $$SPLAST (fetch (SELPIECES SPLAST) of $$SELPIECES))
                                         (SETQ $$SPLENGTH (fetch (SELPIECES SPLEN) of $$SELPIECES))
                                     while I.V. repeatuntil (EQ I.V. $$SPLAST) by (NEXTPIECE I.V.]
       T)
)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'SELECTION
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG
              FULLXPOINTER POINTER POINTER POINTER FLAG POINTER FLAG POINTER)
```

;; ---field descriptor list elided by lister---

```
       '34)

(DEFPRINT 'SELECTION (FUNCTION \TEDIT.SELECTION.DEFPRINT))

(/DECLAREDATATYPE 'SELPIECES '(POINTER POINTER POINTER POINTER POINTER)
```

;; ---field descriptor list elided by lister---

```
       '10)

(DEFINEQ
```

(\TEDIT.SELECTION.DEFPRINT

```
  [LAMBDA (SEL STREAM)                                          ; Edited 11-Feb-2024 08:58 by rmk
                                                                ; Edited  9-Feb-2024 15:55 by rmk
                                                                ; Edited 23-May-2023 00:06 by rmk
                                                                ; Edited 21-May-2023 09:15 by rmk
    (LET ((TEXTOBJ (fetch (SELECTION SELTEXTOBJ) of SEL))
          WHICH INFO LOC)
         (CL:WHEN TEXTOBJ
             (SETQ WHICH (if (EQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
                             then 'SEL
                           elseif (EQ SEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
                             then 'SCRATCH
                           elseif (EQ SEL (fetch (TEXTOBJ SCRATCHSEL2) of TEXTOBJ))
                             then 'SCRATCH2)))
         (SETQ INFO (if (GETSEL SEL SET)
                        then (CONCAT (GETSEL SEL CH#)
                                     "-"
                                     (GETSEL SEL DCH)
                                     "-"
                                     (NTHCHAR (GETSEL SEL POINT)
                                              1)
                                     " "
                                     (CL:IF (EQ (GETSEL SEL HOWHEIGHT)
                                                1)
                                         "_"
                                         (CHARACTER 127)))
                      else "unset"))
         (SETQ LOC (LOC SEL))
         (CONS (CONCAT "{S:" (OR WHICH "?")
                       " " INFO " " (CAR LOC)
                       "/"
                       (CDR LOC)
                       "}"])

)

(DEFINEQ
```

(\TEDIT.SET.GLOBAL.SELECTIONS

```
  [LAMBDA (SELOPERATION SOURCESEL)                              ; Edited 15-Mar-2024 13:38 by rmk
                                                                ; Edited 12-Feb-2024 08:15 by rmk
```

;; This sets the documented global selections (TEDIT.*SELECTION), and some that are not documented (COPYLOOKS, DELETE).

;; SELOPERATION is NIL on loadup, for initialization. Otherwise, SELOPERATION determines which variable is set to a copy of SOURCESEL.

```
    (SELECTQ SELOPERATION
         ((NORMAL PENDINGDEL)
             (SETQ TEDIT.SELECTION (\TEDIT.COPYSEL SOURCESEL)))
         (COPY (SETQ TEDIT.SHIFTEDSELECTION (\TEDIT.COPYSEL SOURCESEL)))
         (MOVE (SETQ TEDIT.MOVESELECTION (\TEDIT.COPYSEL SOURCESEL)))
         (COPYLOOKS (SETQ TEDIT.COPYLOOKSSELECTION (\TEDIT.COPYSEL SOURCESEL)))
         (DELETE (SETQ TEDIT.DELETESELECTION (\TEDIT.COPYSEL SOURCESEL)))
```

```
        (NIL (for S in '(TEDIT.SELECTION TEDIT.SHIFTEDSELECTION TEDIT.COPYLOOKSSELECTION TEDIT.MOVESELECTION
                         TEDIT.DELETESELECTION)
                unless (BOUNDP S) do (SETATOMVAL S (create SELECTION))))
        (SHOULDNT])

)

(\TEDIT.SET.GLOBAL.SELECTIONS)

(DEFINEQ

(\TEDIT.SELECTED.PIECES
  [LAMBDA (TEXTOBJ SEL CROSSCOPY PIECEMAPFN FNARG1 FNARG2)       ; Edited 15-Mar-2024 14:15 by rmk
                                                                 ; Edited 28-Nov-2023 23:14 by rmk
                                                                 ; Edited 21-Jun-2023 20:30 by rmk
                                                                 ; Edited  9-May-2023 13:16 by rmk
                                                                 ; Edited 11-Apr-2023 12:07 by rmk
                                                                 ; Edited 20-Apr-93 17:06 by jds
    ;; NOT SURE THIS IS CALLED

    ;; Create a list of pieces corresponding to the selection;  if FNARG, apply it to each piece, and use the result instead of the piece

    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (CL:UNLESS (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
        (CL:UNLESS SEL
            (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
        (CL:WHEN (GETSEL SEL SET)
            (LET ((SELPIECES (\TEDIT.SELPIECES SEL NIL TEXTOBJ)))
                 (for PC inselpieces (CL:IF CROSSCOPY
                                            (\TEDIT.SELPIECES.COPY SELPIECES 'COPY TEXTOBJ)
                                            SELPIECES)
                    collect (CL:IF PIECEMAPFN
                                   (APPLY* PIECEMAPFN PC TEXTOBJ FNARG1 FNARG2)
                                   PC)))))])

(\TEDIT.FIND.PROTECTED.END
  [LAMBDA (TEXTOBJ CH# LIMITCH#)                                 ; Edited 17-Mar-2024 00:27 by rmk
                                                                 ; Edited  7-Apr-2023 22:13 by rmk
                                                                 ; Edited 23-Oct-2022 17:44 by rmk
                                                                 ; Edited  5-Sep-2022 15:31 by rmk
                                                                 ; Edited 22-Aug-2022 13:21 by rmk
                                                                 ; Edited 18-Apr-93 23:49 by jds
    ;; If LIMITCH# is given, the search will stop there.

    (SETQ LIMITCH# (IMIN LIMITCH# (TEXTLEN TEXTOBJ)))
    (LET (START-OF-PIECE)
         (DECLARE (SPECVARS START-OF-PIECE))
         (for PC inpieces (\TEDIT.CHTOPC CH# TEXTOBJ T) until (IGREATERP START-OF-PIECE LIMITCH#)
            do ;; Move forward thru the pieces of the document, looking for one that contains protected text.  If that comes before the end of the
               ;; region we're interested in, tell the caller about the earlier end to selectable text.
               (CL:WHEN (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC))
                   ;; We've found the beginning of a protected region, previous char is the last selectable.
                   (RETURN (SUB1 START-OF-PIECE)))
               (add START-OF-PIECE (PLEN PC))
            finally (RETURN LIMITCH#])

(\TEDIT.FIND.PROTECTED.START
  [LAMBDA (TEXTOBJ CH# LIMITCH#)                                 ; Edited 17-Mar-2024 00:27 by rmk
                                                                 ; Edited 24-Nov-2023 21:25 by rmk
                                                                 ; Edited  7-Apr-2023 21:59 by rmk
                                                                 ; Edited  4-Feb-2023 10:23 by rmk
                                                                 ; Edited 23-Oct-2022 16:20 by rmk
                                                                 ; Edited  2-Sep-2022 15:26 by rmk
                                                                 ; Edited 22-Aug-2022 13:20 by rmk
                                                                 ; Edited 30-Apr-93 01:39 by jds
    ;; Starting from a CH# in a selectable region, returns the char-number just after the end of the first preceding protected piece.  This is used to limit
    ;; selections to unprotected text, and to prevent selection of the protected text between two unprotected areas.

    ;; Will stop looking when it passes LIMITCH#, or at the beginning of the document.

    (LET (START-OF-PIECE)                                        ; Gets us to the beginning of CH# piece
         (DECLARE (SPECVARS START-OF-PIECE))
         (for PC backpieces (PREVPIECE (\TEDIT.CHTOPC CH# TEXTOBJ T)) until (ILEQ START-OF-PIECE LIMITCH#)
            do (CL:WHEN (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC))
                   ;; Return the CH# just AFTER this first protected piece.
                   (RETURN START-OF-PIECE))
               (add START-OF-PIECE (IMINUS (PLEN PC)))
            finally (RETURN LIMITCH#])

(\TEDIT.WORD.BOUND
  [LAMBDA (TEXTOBJ PREVCH CH)                                    ; Edited 27-Sep-2022 23:54 by rmk
                                                                 ; Edited 25-Sep-2022 23:48 by rmk
```

```
                                                                    ; Edited 30-May-91 23:02 by jds
    (CL:WHEN (AND (FIXP PREVCH)
                  (FIXP CH))
        (LET [(READSA (fetch READSA of (OR (fetch (TEXTOBJ TXTWTBL) of TEXTOBJ)
                                           TEDIT.WORDBOUND.READTABLE]
             (NEQ (\SYNCODE READSA PREVCH)
                  (\SYNCODE READSA CH))))])

)

(RPAQ? TEDIT.EXTEND.PENDING.DELETE T)
```

;; Setting for a "Laurel" mode

;; Selection manipulating code

```
(DEFINEQ
```

(\TEDIT.EXTEND.SEL

```
  [LAMBDA (X Y OSEL TEXTOBJ SELOPERATION PANE)                      ; Edited 15-Mar-2024 13:38 by rmk
                                                                    ; Edited 26-Dec-2023 11:46 by rmk
                                                                    ; Edited 15-Oct-2023 10:39 by rmk
                                                                    ; Edited  5-Oct-2023 22:08 by rmk
                                                                    ; Edited 19-Apr-2023 17:36 by rmk
                                                                    ; Edited 19-Apr-93 13:49 by jds
```

;; Note: CHLIM is one past the last character, hence ADD1/SUB1 in some places below.

;; Obtain a new selection for the character/line that covers X,Y, and extend OSEL to include the additional characters. Return the extended ;; selection.

;; NEWSEL will be a paragraph selection if OSEL was.

;; The paragraph behavior feels a little odd. If you have 4 highlighted paragraphs and click on the 2nd one, the first paragraph is deselected, ;; because the CH# has moved in to the second. If you click on the 3rd paragraph, the last one is deselected (the CHLIM has moved in).

;; Same behavior if you drag from the top or bottom. If you drag from the bottom, the bottom disappears when you enter the 3rd. But as you ;; continue from 3rd to 2nd, the upper deselects. But you might think that dragging behavior would be consistent--as you keep going up (or down), ;; the paragraph you are leaving goes away.

```
    (CL:UNLESS (ZEROP (TEXTLEN TEXTOBJ))
        (PROG (NPOINT NEWSEL OCH# OCHLIM NCH# NCHLIM SETOSELFLG (SELKIND (GETSEL OSEL SELKIND)))
```

;; LINE+WORDSELFLG iff PARA

```
              (SETQ NEWSEL (\TEDIT.SELECT X Y TEXTOBJ (SELECTQ SELKIND
                                                        ((LINE PARA)
                                                            'LINE)
                                                        ((WORD CHAR)
                                                            'TEXT)
                                                        'TEXT)
                                  (OR (EQ SELKIND 'WORD)
                                      (EQ SELKIND 'PARA))
                                  SELOPERATION PANE T))
              (CL:UNLESS (AND NEWSEL (GETSEL NEWSEL SET))           ; No new selection, return OSEL
                  (RETURN OSEL))
              (\TEDIT.SET.SEL.LOOKS NEWSEL SELOPERATION)            ; NEWSEL is the same as OSEL:  regular, copy-source, etc.
              (SETQ OCH# (FGETSEL OSEL CH#))
              (SETQ OCHLIM (FGETSEL OSEL CHLIM))
              (SETQ NCH# (FGETSEL NEWSEL CH#))
              (SETQ NCHLIM (FGETSEL NEWSEL CHLIM))
              [SETQ NPOINT (COND
                             ((IGEQ NCHLIM OCHLIM)                  ; NEWSEL ends to the right of OSEL: adding on the right
                              'RIGHT)
                             ((ILEQ NCH# OCH#)                      ; NEWSEL starts to the left of OSEL: adding on the left
                              'LEFT)
                             ((IGREATERP (IABS (IDIFFERENCE NCHLIM OCHLIM))
                                     (IABS (IDIFFERENCE NCH# OCH#)))
                              ;; New X (right click) is in the middle of an old selection.  Must be shrinking from the left. This determines the
                              ;; relationships based on character positions.  It might be more intuitive in PARA mode if this is based on
                              ;; paragraphs--if there are fewer *paragraphs* in front than behind, of any length.
                              (SETQ SETOSELFLG T)
                              'LEFT)
                             (T ;; Must be shrinking from the right.  Move OLDSEL's CHLIM back to NEWSEL's
                                (SETQ SETOSELFLG T)
                                'RIGHT]
              (SELECTQ NPOINT
                  (LEFT                                             ; Caret's to the new left, keep old right
                        (SETQ NCHLIM (IMAX NCHLIM OCHLIM))
                        (FSETSEL NEWSEL CHLIM NCHLIM)
                        (FSETSEL NEWSEL XLIM (FGETSEL OSEL XLIM))
                        (FSETSEL NEWSEL LN (COPY (FGETSEL OSEL LN)))
                        ;; Only copying is allowed from a protected area (menu).  Otherwise, only extend to its start. If CH# changes, L1 may
                        ;; also change (\FIXSEL)
                        (CL:UNLESS (EQ SELOPERATION 'COPY)
                            (SETQ NCH# (IMAX NCH# (\TEDIT.FIND.PROTECTED.START TEXTOBJ (SUB1 OCHLIM)
                                                    NCH#)))))
                  (RIGHT                                            ; Caret's to the new right, keep old left
```

```
                        (SETQ NCH# (IMIN NCH# OCH#))
                        (FSETSEL NEWSEL X0 (FGETSEL OSEL X0))
                        (FSETSEL NEWSEL L1 (COPY (FGETSEL OSEL L1)))
                        ;; Only copying is allowed from a protected area (menu).  Otherwise, only extend to its end. If CHLIM changes, LN
                        ;; may also change (\FIXSEL)
                        (CL:UNLESS (EQ SELOPERATION 'COPY)
                            [SETQ NCHLIM (IMIN NCHLIM (ADD1 (\TEDIT.FIND.PROTECTED.END
                                                              TEXTOBJ OCH# (ADD1 (\TEDIT.FIND.PROTECTED.END
                                                                                   TEXTOBJ OCH# (SUB1 NCHLIM]
                            (SETQ NCH# (IMIN NCH# (SUB1 NCHLIM)))))
                    (SHOULDNT))
                (FSETSEL NEWSEL CH# NCH#)
                (FSETSEL NEWSEL CHLIM NCHLIM)
                (FSETSEL NEWSEL DCH (IDIFFERENCE NCHLIM (FGETSEL NEWSEL CH#)))
                (FSETSEL NEWSEL POINT NPOINT)
                (CL:UNLESS (EQ (FGETSEL OSEL SELOBJ)
                               (FGETSEL NEWSEL SELOBJ))            ; Keep object if it is in overlapping part?
                    (FSETSEL NEWSEL SELOBJ NIL))
                (\TEDIT.FIXSEL NEWSEL TEXTOBJ)
                (CL:WHEN SETOSELFLG                                ; It is wise to copy the new sel into the old one.
                    (\TEDIT.COPYSEL NEWSEL OSEL))
                (RETURN NEWSEL)))])
```

(\TEDIT.SELECT

```
  [LAMBDA (X Y TEXTOBJ REGION WORDSELFLG SELOPERATION PANE EXTENDING)
                                                                   ; Edited 15-Mar-2024 13:36 by rmk
                                                                   ; Edited  2-Jan-2024 12:32 by rmk
                                                                   ; Edited 26-Dec-2023 08:50 by rmk
                                                                   ; Edited 23-May-2023 12:38 by rmk
                                                                   ; Edited  9-Apr-2023 23:01 by rmk
                                                                   ; Edited 30-May-91 23:07 by jds
    ;; Select the character word, line, or paragraph the mouse is pointing at.
    (LET ((SEL (\TEDIT.SELECT.LINE.SCANNER X Y TEXTOBJ (fetch (TEXTWINDOW PLINES) of PANE)
                      REGION WORDSELFLG SELOPERATION PANE EXTENDING)))
         (CL:WHEN (AND (type? SELECTION SEL)
                       (GETSEL SEL SET))                           ; He pointed at something real;  return that.
             (\TEDIT.SET.SEL.LOOKS SEL SELOPERATION)
             (\TEDIT.FIXSEL SEL TEXTOBJ PANE)                      ; This PANE is good, fix all the other ones
            SEL)])
```

(\TEDIT.SCAN.LINE

```
  [LAMBDA (TEXTOBJ LINE X Y WORDSELFLG SELOPERATION PANE EXTENDING)
                                                                   ; Edited 15-Mar-2024 19:22 by rmk
                                                                   ; Edited 27-Jan-2024 23:44 by rmk
                                                                   ; Edited 26-Jan-2024 21:54 by rmk
                                                                   ; Edited 22-Jan-2024 17:15 by rmk
                                                                   ; Edited  3-Jan-2024 00:34 by rmk
                                                                   ; Edited 14-Oct-2023 10:46 by rmk
                                                                   ; Edited  5-May-2023 00:18 by rmk
                                                                   ; Edited 18-Apr-2023 23:09 by rmk
                                                                   ; Edited  9-Apr-2023 18:21 by rmk
                                                                   ; Edited 31-May-91 12:26 by jds
    ;; Given that LINE meets the mouse-Y criterion, find the selection picked out by the mouse X  coordinate.  This may run to the right if the
    ;; mouse-position is protected.
    (PROG (SCRSEL CHARSLOT CLOOKS CHNO TXB TX SELSLOT SELCHAR SELHERE PASTRIGHT (THISLINE (FGETTOBJ TEXTOBJ
                                                                                                THISLINE)))
          (CL:UNLESS (EQ LINE (fetch DESC of THISLINE))            ; Make sure the cache describes this line
              (SETQ LINE (\TEDIT.FORMATLINE TEXTOBJ (GETLD LINE LCHAR1)
                                LINE)))                            ; Convert X's display units to LINE's scale
          (SETQ TX (GETLD LINE LX1))
          (SETQ TXB TX)
          (SETQ X (IMAX X TX))                                     ; Move over if the click was in the left margin.
          (SETQ CHNO (FGETLD LINE LCHAR1))
     ;;
     ;; Step 1:  Find the slot, character number, and ending TX for the character at the incoming mouse X position.
          (CL:WHEN (SETQ PASTRIGHT (IGREATERP X (FGETLD LINE LXLIM)))
                                                                   ; Past the end, put it inside the last character
              (SETQ X (SUB1 (FGETLD LINE LXLIM))))
          (for old CHARSLOT incharslots THISLINE do (CL:UNLESS CHAR
                                                                   ; Invisible or charlooks
                                                        (CL:IF (SMALLP CHARW)
                                                            (add CHNO CHARW)
                                                            (SETQ CLOOKS CHARW))
                                                        (GO $$ITERATE))
                                                    (SETQ TXB TX)
                                                    (add TX CHARW)
                                                    (CL:WHEN (IGEQ TX X)
                                                        (if SELHERE
                                                            then (SETQ TX TXB)
                                                                   ; Now presumaby looking at }, we want a 0-char selection at TXB
                                                                 (SETQ X TX)
```

```
                                                    (RETURN)
                                                else
                                                    ;; Presumably the end of a selected region in a menu, probably {.  It
                                                    ;; appears that we want to go one more
                                                      (SETQ SELHERE (fetch (CHARLOOKS CLSELHERE)
                                                                       of CLOOKS)))
                                                (CL:UNLESS (fetch (CHARLOOKS CLPROTECTED) of CLOOKS)
                                                        ; If protected, we keep going beyond the given X
                                                    (RETURN)))
                                            (add CHNO 1)
        finally ;; We lose if all characters after X are protected.
                (CL:WHEN (fetch (CHARLOOKS CLPROTECTED) of CLOOKS)
                        (SETQ CHARSLOT NIL)))
    (CL:UNLESS CHARSLOT                                     ; Everything after X was protected.
        (RETURN 'DON'T))
;; CHNO and CHARSLOT:  the character pointed to, CLOOKS the looks of that character.
;; CHNO and CHARSLOT are either flagged as CLSELHERE or are not flagged as CLPROTECTED.
;; TXB the end of CHNO-1, TX the end of CHNO.  They both may be beyond X, if protected.
;;
    (SETQ SELSLOT CHARSLOT)
    (SETQ SELCHAR (CHAR SELSLOT))
;;  CHNO and SELSLOT define a selectable character/object SELCHAR that runs from TXB to TX.
;;
;; The selection runs from TXB to TX and from CHNOB to CHNO.  In the character case, CHNOB=CHNO and TX-TB is the selection width (DX).
;; If the selected piece is SELHERE (presumably in a menu), the selection is specialized in various ways..
    (SETQ SCRSEL (FGETTOBJ TEXTOBJ SCRATCHSEL))
    (FSETSEL SCRSEL SELTEXTOBJ TEXTOBJ)
    (FSETSEL SCRSEL SET T)
    (FSETSEL SCRSEL SELKIND 'CHAR)
    (FSETSEL SCRSEL X0 TXB)                                 ; X and Y values will be reset by \FIXSEL, but we have to track X
                                                            ; for word selection and image obj
    (FSETSEL SCRSEL XLIM TX)
    (FSETSEL SCRSEL CH# CHNO)
    (FSETSEL SCRSEL CHLIM (ADD1 CHNO))
    (FSETSEL SCRSEL SELOBJ NIL)
;; 0 makes it a point selection, 1 picks out a single char.  Original code produced 0 only for protected text and dummy lines.
    (FSETSEL SCRSEL DCH (CL:IF (AND (EQ SELOPERATION 'NORMAL)
                                    (NOT (FGETTOBJ TEXTOBJ TXTREADONLY)))
                            0
                            1))
    (FSETSEL SCRSEL POINT (if (OR SELHERE (AND PASTRIGHT (FGETLD LINE FORCED-END)))
                              then
                                  ;; This is coordinated with the point selection in \FIXSEL. If we are past the end of an EOL-line, we
                                  ;; want the caret to blink on the left but select and underline the EOL.
                                  'LEFT
                              elseif (AND (IGEQ (IDIFFERENCE TX TXB)
                                                3)
                                          (IGEQ X (FOLDLO (IPLUS TX TXB)
                                                          2)))
                              then
                                  ;; To the right of an otherwise-protected insertion, past the middle of a selection that is wide enough
                                  ;; (3 points) to discriminate, and not at the  end of an EOL-terminated line.
                                  'RIGHT
                              else 'LEFT))
    (CL:WHEN (AND WORDSELFLG (NOT (FGETLD LINE LDUMMY)))
        ;; Expand the selection to its word boundaries
        (\TEDIT.SCAN.LINE.WORD X TEXTOBJ THISLINE SCRSEL SELSLOT CLOOKS))
    (CL:WHEN (AND (type? IMAGEOBJ SELCHAR)
                  (NOT PASTRIGHT))                          ; Don't interpret an object that X was backed up to.
        (\TEDIT.SELECT.OBJECT TEXTOBJ SCRSEL SELCHAR LINE X Y TXB PANE SELOPERATION (COND
                                                                                        (EXTENDING
                                                                                         'RIGHT)
                                                                                        (WORDSELFLG
                                                                                         'MIDDLE)
                                                                                        (T 'LEFT))
                EXTENDING))
    (for L1 on (FGETSEL SCRSEL L1) as LN on (FGETSEL SCRSEL LN) as P inpanes TEXTOBJ
       when (EQ P PANE) do (RPLACA L1 LINE)
                           (RPLACA LN LINE))
    (RETURN SCRSEL])
```

(\TEDIT.SCAN.LINE.WORD

```
  [LAMBDA (X TEXTOBJ THISLINE SCRSEL SELSLOT SELLOOKS)          ; Edited 24-Dec-2023 22:04 by rmk
                                                                ; Edited 14-Oct-2023 10:33 by rmk
                                                                ; Edited 26-May-2023 23:05 by rmk
```

```
                                                                    ; Edited 20-Mar-2023 23:42 by rmk
                                                                    ; Edited  6-Mar-2023 22:22 by rmk
                                                                    ; Edited  2-Mar-2023 14:56 by rmk
                                                                    ; Edited 26-Feb-2023 15:55 by rmk
```

;; SCRSEL is a character selection at the SELSLOT character in THISLINE. This expands it to its surrounding word boundaries. Looks are tracked ;; for protection.

;;

```
(\DTEST SCRSEL 'SELECTION)
(CL:UNLESS (EQ 'CHAR (FGETSEL SCRSEL SELKIND))
       (SHOULDNT "Can only expand CHAR selections to WORD selections"))
(LET (CH# CHLIM X0 XLIM)
```

;; CH# will be the first charno of the word selection

;; CHLIM will be one past the last charno of the word selection

;; X0 will be the X at the beginning of the first char

;; XLIM will be the X at the end of last charL

```
(SETQ CH# (FGETSEL SCRSEL CH#))
(SETQ CHLIM (FGETSEL SCRSEL CHLIM))                    ;
(SETQ X0 (FGETSEL SCRSEL X0))
(SETQ XLIM (FGETSEL SCRSEL XLIM))
(for CHARSLOT (CLOOKS _ SELLOOKS)
    (LASTCHAR _ (CHAR SELSLOT))
    backcharslots
    (PREVCHARSLOT SELSLOT) do (CL:UNLESS CHAR
                                  (SETQ CLOOKS CHARW)
                                  (GO $$ITERATE))
                              (CL:WHEN (OR (type? IMAGEOBJ CHAR)
                                           (\TEDIT.WORD.BOUND TEXTOBJ CHAR LASTCHAR)
                                           (fetch (CHARLOOKS CLPROTECTED) of CLOOKS))
                                                    ; Stop at a protection bounary
                                  (RETURN))
                              (SETQ LASTCHAR CHAR)
                              (ADD X0 (IMINUS CHARW))
                              (ADD CH# -1))
```

;; And search forward for the end of the word

```
(for CHARSLOT (CLOOKS _ SELLOOKS)
    (PREVCHAR _ (CHAR SELSLOT))
    incharslots
    (NEXTCHARSLOT SELSLOT) do (CL:UNLESS CHAR
                                  (SETQ CLOOKS CHARW)
                                  (GO $$ITERATE))
                              (CL:WHEN (OR (type? IMAGEOBJ CHAR)
                                           (\TEDIT.WORD.BOUND TEXTOBJ PREVCHAR CHAR)
                                           (fetch (CHARLOOKS CLPROTECTED) of CLOOKS))
```

;; XLIM is now the end of the last character of the word.

;; CHLIM and XLIM should be OK if we run off the end.

```
                                  (RETURN))
                              (add XLIM CHARW)
                              (add CHLIM 1)
                              (SETQ PREVCHAR CHAR))
(FSETSEL SCRSEL SELKIND 'WORD)
(FSETSEL SCRSEL CH# CH#)
(FSETSEL SCRSEL CHLIM CHLIM)
(FSETSEL SCRSEL DCH (IDIFFERENCE CHLIM CH#))
(FSETSEL SCRSEL X0 X0)
(FSETSEL SCRSEL XLIM XLIM)
```

;; Move the point to the intended side of the word: To the right of an otherwise-protected insertion, past the middle of a selection that is wide ;; enough to discriminate, and not at the end of an EOL-terminated line. 3 is points.

```
(FSETSEL SCRSEL POINT (if [OR (fetch (CHARLOOKS CLSELHERE) of SELLOOKS)
                              (AND (IGEQ (IDIFFERENCE XLIM X0)
                                         3)
                                   (IGEQ X (FOLDLO (IPLUS XLIM X0)
                                                   2]
                         then 'RIGHT
                        else 'LEFT])
```

(**TEDIT.SELECT.LINE.SCANNER**

```
  [LAMBDA (X Y TEXTOBJ LINES REGION WORDSELFLG SELOPERATION PANE EXTENDING)
                                                                    ; Edited 15-Mar-2024 13:36 by rmk
                                                                    ; Edited 26-Dec-2023 08:53 by rmk
                                                                    ; Edited  3-Nov-2023 12:00 by rmk
                                                                    ; Edited 14-Oct-2023 22:43 by rmk
                                                                    ; Edited 20-Jul-2023 20:38 by rmk
                                                                    ; Edited 30-May-2023 14:17 by rmk
                                                                    ; Edited 27-May-2023 15:18 by rmk
                                                                    ; Edited 31-May-91 12:26 by jds
   (CL:WHEN (INSIDEP (DSPCLIPPINGREGION NIL PANE)
                     X Y)                                           ; Else, how did we get here?
       (PROG (LINE SCRSEL PARAFIRSTCHNO PARALASTCHNO)
             [SETQ LINE (find L PREV inlines (GETLD LINES NEXTLINE) suchthat (SETQ PREV (FGETLD L PREVLINE))
```

```
                                                                (ILEQ (FGETLD L YBOT)
                                                                      Y)
                finally ;; Y is below thelast line.  Assume it points to the last.
                        (RETURN (OR L PREV]
        (CL:UNLESS LINE                                 ; Can this happen?  Empty?
            (RETURN NIL))
        (SELECTQ REGION
            ((TEXT PANE)                                ; We're in the regular text area, which character?
                (CL:WHEN (AND (IGEQ (GETLD LINE LCHARLIM)
                                    (TEXTLEN TEXTOBJ))
                              (IGREATERP (GETLD LINE YBOT)
                                         Y))
                    ;; Y is below the last line of the text: force selection past the very end of that line.
                    (SETQ X (ADD1 (GETLD LINE LXLIM))))
                (RETURN (\TEDIT.SCAN.LINE TEXTOBJ LINE X Y WORDSELFLG SELOPERATION PANE EXTENDING)))
            (LINE                                       ; We're in the line/paragraph region.
                (SETQ SCRSEL (FGETTOBJ TEXTOBJ SCRATCHSEL))
                (CL:WHEN (AND (GETLD LINE LHASPROT)
                              (NEQ SELOPERATION 'COPY))
                    ;; In a TEDIT menu, you can't select a whole paragraph or line.
                    (FSETSEL SCRSEL SET NIL)
                    (RETURN SCRSEL))
                (FSETSEL SCRSEL SELTEXTOBJ TEXTOBJ)
                (FSETSEL SCRSEL SET T)                  ; Mark it valid.
                (FSETSEL SCRSEL SELOBJ NIL)             ; Not selecting an object just yet
                ;; Get the lines selected in this pane.  How does SCRATCHSEL know this?
                (for P inpanes TEXTOBJ as PL1 on (FGETSEL SCRSEL L1) as PLN on (FGETSEL SCRSEL LN)
                   when (EQ P PANE) do ;; A word (middle button?) selection in the line region means the paragraph that
                                       ;; contains the selected line.
                                       (if WORDSELFLG
                                           then
                                       ;; We have to find its first and last character numbers, whether or not they are visible in any pane.
                                       ;; \FIXSEL will figure out the (sub?) set of lines that are visible in this pane, other panes are done at a
                                       ;; higher level
                                            (SETQ PARAFIRSTCHNO (CAR (\TEDIT.PARA.FIRST TEXTOBJ
                                                                                        (FGETLD LINE LCHAR1)
                                                                                        T)))
                                            (SETQ PARALASTCHNO (CAR (\TEDIT.PARA.LAST TEXTOBJ
                                                                                      (FGETLD LINE LCHARLIM)
                                                                                      T)))
                                       ;; If LINE is closer to the beginning of the paragraph, put the point before the first line.
                                       ;; Otherwise after the last line.
                                            (\TEDIT.UPDATE.SEL SCRSEL PARAFIRSTCHNO
                                                   (IDIFFERENCE (ADD1 PARALASTCHNO)
                                                          PARAFIRSTCHNO)
                                                   (COND
                                                      ((ILEQ (IDIFFERENCE (FGETLD LINE LCHAR1)
                                                                          PARAFIRSTCHNO)
                                                             (IDIFFERENCE (ADD1 PARALASTCHNO)
                                                                    (FGETLD LINE LCHARLIM)))
                                                       'LEFT)
                                                      (T 'RIGHT))
                                                   NIL T)
                                            (FSETSEL SCRSEL SELKIND 'PARA)
                                            (\TEDIT.FIXSEL SCRSEL TEXTOBJ NIL PANE)
                                       else                ; Select just the line we're pointing at.
                                            (RPLACA PL1 LINE)
                                            (RPLACA PLN LINE)
                                            (FSETSEL SCRSEL SELKIND 'LINE)
                                            (FSETSEL SCRSEL SET T)
                                            (\TEDIT.UPDATE.SEL SCRSEL (FGETLD LINE LCHAR1)
                                                   (IDIFFERENCE (ADD1 (FGETLD LINE LCHARLIM))
                                                          (FGETLD LINE LCHAR1))
                                                   'LEFT NIL T)
                                       ;; In the line-selection region, we know that the selection's X0 and XLIM are
                                       ;; inherited from the LINE. Don't need to fix
                                            (FSETSEL SCRSEL X0 (FGETLD LINE LX1))
                                            (FSETSEL SCRSEL XLIM (FGETLD LINE LXLIM)))
                   finally (RETURN))
                (RETURN SCRSEL))
            (SHOULDNT "Unknown text/line-bar region?")))])
```

(**TEDIT.SELECT.OBJECT**

```
  [LAMBDA (TEXTOBJ SEL OBJ LINE X Y TXB SELPANE SELOPERATION WHERE)
                                                                ; Edited 15-Mar-2024 19:22 by rmk
                                                                ; Edited 24-Jan-2024 11:59 by rmk
                                                                ; Edited 14-Oct-2023 11:38 by rmk
                                                                ; Edited 10-Apr-2023 08:38 by rmk
```

```
                                                                    ; Edited 29-Mar-94 13:28 by jds
    (SETSEL SEL SELOBJ OBJ)
    (SETSEL SEL X0 TXB)
    (CL:WHEN (AND (EQ WHERE 'LEFT)
                  (EQ (FGETSEL SEL DCH)
                      0))
           (FSETSEL SEL DCH 1))
    (LET ([OBJBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
                      (IMAGEBOX OBJ SELPANE 'DISPLAY]
          (DS (WINDOWPROP SELPANE 'DSP))
          SELRES)
         (RESETLST
             (RESETSAVE (DSPXOFFSET (IDIFFERENCE (IPLUS TXB (DSPXOFFSET NIL DS))
                                                 (fetch XKERN of OBJBOX))
                                    DS)
                    (LIST (FUNCTION DSPXOFFSET)
                          (DSPXOFFSET NIL DS)
                          DS))
             (RESETSAVE (DSPYOFFSET (IDIFFERENCE (IPLUS (GETLD LINE YBASE)
                                                        (DSPYOFFSET NIL DS))
                                                 (fetch YDESC of OBJBOX))
                                    DS)
                    (LIST (FUNCTION DSPYOFFSET)
                          (DSPYOFFSET NIL DS)
                          DS))
             (RESETSAVE (DSPCLIPPINGREGION (create REGION
                                                   LEFT _ 0
                                                   BOTTOM _ 0
                                                   WIDTH _ (IMIN (fetch XSIZE of OBJBOX)
                                                                 (IDIFFERENCE (FGETTOBJ TEXTOBJ WRIGHT)
                                                                              TXB))
                                                   HEIGHT _ (fetch YSIZE of OBJBOX))
                                    DS)
                    (LIST (FUNCTION DSPCLIPPINGREGION)
                          (DSPCLIPPINGREGION NIL DS)
                          DS))                                      ; Go tell him he's being pointed at.
             (SETQ SELRES (ERSETQ (APPLY* (IMAGEOBJPROP OBJ 'BUTTONEVENTINFN)
                                          OBJ DS SEL (IDIFFERENCE X TXB)
                                          (IDIFFERENCE Y (GETLD LINE YBASE))
                                          SELPANE
                                          (FGETTOBJ TEXTOBJ STREAMHINT)
                                          WHERE SELOPERATION))))
          ;; The clipping region is now restored.
          (CL:WHEN (LISTP SELRES)                                   ; If not a LIST, an ereror happened
              (SELECTQ (CAR SELRES)
                   (NIL                                             ; Do nothing untoward
                        (FSETSEL SEL SELOBJ NIL))
                   (DON'T                                           ; The object declines to be selected.
                          (FSETSEL SEL SET NIL))
                   (CHANGED                                         ; Update the screen
                            (\TEDIT.FORMATLINE TEXTOBJ (GETLD LINE LCHAR1)
                                   LINE)
                            (\TEDIT.DISPLAYLINE TEXTOBJ LINE SELPANE)
                            (TEDIT.OBJECT.CHANGED TEXTOBJ (fetch (SELECTION SELOBJ) of SEL)))
                   NIL))])

)

(DEFINEQ

(\TEDIT.FIXSEL
  [LAMBDA (SEL TEXTOBJ AVOIDPANE ONLYPANE)                          ; Edited 20-Mar-2024 10:55 by rmk
                                                                    ; Edited  2-Mar-2024 23:38 by rmk
                                                                    ; Edited 16-Dec-2023 11:44 by rmk
                                                                    ; Edited  3-Nov-2023 12:01 by rmk
                                                                    ; Edited 28-Jul-2023 15:58 by rmk
                                                                    ; Edited 22-Jun-2023 16:05 by rmk
                                                                    ; Edited  6-Jun-2023 13:26 by rmk
                                                                    ; Edited  1-Jun-2023 17:41 by rmk
                                                                    ; Edited 31-May-91 12:26 by jds
    ;; PLINES of each PANE heads the list of lines that are visible in that pane. This routine determines which of those visible lines contains characters
    ;; between the first and last characters that are selected by SEL, if any.  The first visible and selected line is stored in the L1 component of the
    ;; selection that corresponds to PANE, and the last visible/selected line is stored in the LN. L1 and LN can both either be NIL (selection is not
    ;; visible in a pane) or both be lines (if the pane shows a startiing selected line, it must necessarily show an ending line).

    ;;

    ;; If the first selected line in a pane is the line containing the first character of the selection, then X0 is calculated for the selection.  Since panes are
    ;; all the same width, the X0 is the same for all panes in which the first selected line is visible, and so is computed only once.  XLIM is similarly
    ;; calculated if the last character of the selection is visible in a pane.  X0 and XLIM values are irrelevant (and may remain NIL) if the first/last lines
    ;; are not visible in any pane.

    ;;

    ;; Selections also used to contain  starting and ending Y values, but those are pane-dependent and no longer made sense once multiple panes
    ;; were introduced.

    ;;
```

```
;; AVOIDPANE is provided for a pane that may be skipped, e.g. the current selection pane.  Its properties are already known, no point in doing
;; extra work.

;; ONLYPANE is specified in scrolling. to avoid disturbing and redisplaying panes that are not been scrolled.

;;

;; Assumes that the per-pane lines are properly broken so that a forced-end selection can safely move to the next line (after an EOL insertion).

;;

;; Selection L1 and LN are sequences of CONS cells one for each pane that the text appears in. The running (CAR L1) heads the sub-chain of lines
;; selected for the current pane, the running (CAR LN) points to the pane's last selected line.

;;

;; Each pane's PLINES is a constant (dummy) line somewhere previous to the first visible line in that pane.

;;

;; If TXTDON'TUPDATE, the lines may not correspond to anything reasonable, don't try to find X.

(\DTEST SEL 'SELECTION)
(CL:UNLESS TEXTOBJ
    (SETQ TEXTOBJ (GETSEL SEL SELTEXTOBJ)))
(TEXTOBJ! TEXTOBJ)
(CL:WHEN (AND (FGETTOBJ TEXTOBJ \WINDOW)
              (FGETSEL SEL SET)
              (NOT (FGETTOBJ TEXTOBJ TXTDON'TUPDATE)))
    ;; CH# is the first selected character, CHLIM is the character just after the last one, hence the SUB1.

    ;; For a point selection, CHLIM=(ADD1 CH#) so CHNO=LASTCHNO, and the caret position is determined by POINT.  Highlighting is
    ;; determined separately by DCH, which is 0 for point selections.

    (for PANE PSTARTLINE PENDLINE X0 XLIM (CH# _ (IMAX 1 (FGETSEL SEL CH#)))
        [LASTCHNO _ (IMAX 1 (SUB1 (FGETSEL SEL CHLIM]
        inpanes TEXTOBJ as L1 on (FGETSEL SEL L1) as LN on (FGETSEL SEL LN) unless (EQ PANE AVOIDPANE)
      when (OR (NULL ONLYPANE)
               (EQ PANE ONLYPANE))
      when (SETQ PSTARTLINE (find L inlines (GETLD (fetch (TEXTWINDOW PLINES) of PANE)
                                                   NEXTLINE)
                                  suchthat
                                      ;; The first visible line in PANE that contains or follows CHNO.
                                      (LINESELECTEDP L CH# LASTCHNO)
                                  finally ;; Suchthat always comes here: start by asserting no visible lines,  $$VAL=NIL if no visible lines
                                          ;; in this pane
                                      (RPLACA L1 NIL)
                                      (RPLACA LN NIL)))
      do [if (EQ 0 (FGETSEL SEL DCH))
             then ;; Point selection, CHNO=LASTCHNO, POINT determines whether the caret blinks before or after that character.
                  (CL:WHEN (AND (FGETLD PSTARTLINE FORCED-END)
                                (IEQP CH# (FGETLD PSTARTLINE LCHARLIM))
                                (EQ 'RIGHT (FGETSEL SEL POINT))
                                (FGETLD PSTARTLINE NEXTLINE))
                      ;; Point to the right of the EOL that forced a line.  Advance to the beginning of the next line..
                      (SETQ PSTARTLINE (FGETLD PSTARTLINE NEXTLINE))
                      (SETQ CH# (FGETLD PSTARTLINE LCHAR1))
                      (SETQ LASTCHNO CH#)
                      (FSETSEL SEL CH# CH#)
                      (FSETSEL SEL CHLIM (ADD1 CH#))
                      (FSETSEL SEL POINT 'LEFT))
                  (SETQ PENDLINE PSTARTLINE)
                  (CL:UNLESS X0                                ; May have been computed for a prior pane
                      (CL:WHEN (WITHINLINEP CH# PSTARTLINE)
                          [SETQ X0 (\TEDIT.CHTOX TEXTOBJ PSTARTLINE CH# (EQ 'RIGHT (FGETSEL SEL POINT]
                          (FSETSEL SEL X0 X0)
                          (FSETSEL SEL XLIM X0)))
         else ;; For highlighting, if the PSTARTLINE for PANE is also the first line of the selection, then update the selection's X0. Similarly
              ;; for XLIM and PENDLINE. \SHOWSEL.HILIGHT uses the l LX1 and LXLIM values for interior lines. (Except:  If LASTCHNO
              ;; is after a text-final EOL, X0 is the right-edge.)
              [SETQ PENDLINE (for L (PBOTTOM _ (fetch (REGION BOTTOM) of (DSPCLIPPINGREGION NIL PANE)))
                                inlines PSTARTLINE do ;; Stop when L is beyond  the selection or below the screen.
                                                      (CL:WHEN (ILEQ LASTCHNO (FGETLD L LCHARLIM))
                                                            (RETURN L))
                                                      (CL:WHEN (ILEQ (FGETLD L YBOT)
                                                                     PBOTTOM)
                                                          ; This can happen if LASTCHAR is not visible on the screen
                                                          (RETURN $$PREVLINE))
                                    finally ;; If $$PREVLINE is NIL, we didn't advance--so we must have ended at the start
                                            (RETURN (OR $$PREVLINE PSTARTLINE]
              (CL:UNLESS PENDLINE                              ; Start could be the last line in the window, it ends there too.
                  (SETQ PENDLINE PSTARTLINE))
              ;; IMAX to use the first character of PSTARTLINE if it is not the first line of the selection
              (CL:UNLESS X0                                    ; May have been computed for a prior pane
```

```
                    (CL:WHEN (WITHINLINEP CH# PSTARTLINE)
                        [SETQ X0 (\TEDIT.CHTOX TEXTOBJ PSTARTLINE (IMAX CH# (FGETLD PSTARTLINE LCHAR1))
                                       (AND (IGREATERP CH# (TEXTLEN TEXTOBJ))
                                            (GETLD (FGETLD PSTARTLINE PREVLINE)
                                                   FORCED-END]
                        (FSETSEL SEL X0 X0)))
                ;; IMIN to use the last character of PENDLINE if it is not the last line of the selection
                (CL:UNLESS XLIM
                    (CL:WHEN (WITHINLINEP LASTCHNO PENDLINE)
                        (SETQ XLIM (\TEDIT.CHTOX TEXTOBJ PENDLINE LASTCHNO T))
                        (FSETSEL SEL XLIM XLIM)))]
            ;; Fill in the selection
            (RPLACA L1 PSTARTLINE)
            (RPLACA LN PENDLINE)))
    SEL])
```

(\TEDIT.CHTOX

```
  [LAMBDA (TEXTOBJ LINE CH# AFTER)                                    ; Edited 15-Mar-2024 19:22 by rmk
                                                                      ; Edited 23-Dec-2023 14:07 by rmk
                                                                      ; Edited  2-Dec-2023 10:01 by rmk
                                                                      ; Edited 16-May-2023 00:20 by rmk
                                                                      ; Edited 23-Mar-2023 23:04 by rmk
```

;; Return the screen-point X position of character CH# in LINE.

;; If AFTER, returns the Xposition at the end of CH#, otherwise at the beginning.

;; it is an error if CH# is before LCHAR1 or after LCHARLIM.

```
    (\DTEST LINE 'LINEDESCRIPTOR)
    (LET (X (THISLINE (GETTOBJ TEXTOBJ THISLINE)))
         (CL:WHEN (OR (FGETLD LINE LDIRTY)
                      (NEQ LINE (fetch DESC of THISLINE)))
```

;; Reformat if LINE is dirty or not cached in THISLINE.

```
             (\TEDIT.FORMATLINE TEXTOBJ (FGETLD LINE LCHAR1)
                    LINE))
```

;; Can avoid another loop if we are asking about the first or last characters.

```
         (if (AND AFTER (IEQP CH# (FGETLD LINE LCHARLIM)))
             then (FGETLD LINE LXLIM)
           elseif (AND (NOT AFTER)
                       (IEQP CH# (FGETLD LINE LCHAR1)))
             then (FGETLD LINE LX1)
           else (for CHARSLOT (X _ (FGETLD LINE LX1))
                     (CHNO _ (FGETLD LINE LCHAR1))
                     incharslots THISLINE unless (type? CHARLOOKS CHARW)
                   do ;; Update the running X-position in the line, skiping look-slots
                      (CL:WHEN (IEQP CHNO CH#)
                          (if AFTER
                              then (add X (CHARW CHARSLOT)))
                      ;; Scale selection X down to points for lines in hardcopy-display mode.
                          (RETURN X))
                      (CL:WHEN CHAR                                   ; Ignore CHARLOOKS
                          (add CHNO 1)
                          (add X CHARW))
                   finally (CL:WHEN (AND (IEQP CH# (FGETLD LINE LCHAR1))
                                         (IGEQ (FGETLD LINE LCHARLIM)
                                               (FGETTOBJ TEXTOBJ TEXTLEN))
                                         (EQ (FGETLD LINE LXLIM)
                                             (FGETLD LINE LX1)))
                              ;; CH# not found in empty final line, return left margin
                              (RETURN (FGETLD LINE LX1)))])
```

(\TEDIT.COLLECTSELS

```
  [LAMBDA (TEXTOBJ AVOIDSEL)                                          ; Edited 20-Mar-2024 10:56 by rmk
                                                                      ; Edited 11-Feb-2024 09:21 by rmk
                                                                      ; Edited  9-Feb-2024 15:55 by rmk
                                                                      ; Edited 20-Sep-2023 17:02 by rmk
                                                                      ; Edited  9-Sep-2023 17:15 by rmk
                                                                      ; Edited 26-Mar-2023 20:30 by rmk
                                                                      ; Edited 30-May-91 23:03 by jds
```

;; AVOIDSEL to avoid double hits on selections that we might be dealing with separately (e.g. SCRATCHSEL) MAYBE NOT USED

```
    (TEXTOBJ! TEXTOBJ)
    (DREMOVE AVOIDSEL (DREMOVE NIL (LIST (FGETTOBJ TEXTOBJ SEL)
                                         (FGETTOBJ TEXTOBJ SCRATCHSEL)
                                         (FGETTOBJ TEXTOBJ SCRATCHSEL2])
```

(\TEDIT.SELECTION.UNSET

```
  [LAMBDA (SEL)                                                       ; Edited 23-May-2023 13:52 by rmk
```

```
    ;; Unsets a selection, wiping out things that are no longer needed and might be confusing
    (SETSEL SEL SET NIL)
    (SETSEL SEL L1 NIL)
    (SETSEL SEL LN NIL])

)

(DEFINEQ
```

(\TEDIT.RESET.EXTEND.PENDING.DELETE

```
  [LAMBDA (SEL TEXTOBJ)                                    ; Edited  9-Mar-2024 11:37 by rmk
                                                           ; Edited 19-Feb-2024 23:10 by rmk
                                                           ; Edited 24-Dec-2023 00:18 by rmk
                                                           ; Edited  4-May-2023 00:08 by rmk
                                                           ; Edited 21-Oct-2022 18:41 by rmk
                                                           ; Edited 30-May-91 23:03 by jds
    ;; Reset the 'Extend Pending Delete' status
    (CL:WHEN SEL
        (\TEDIT.SET.SEL.LOOKS SEL 'NORMAL)
        (SETTOBJ TEXTOBJ BLUEPENDINGDELETE NIL))])
```

(\TEDIT.SET.SEL.LOOKS

```
  [LAMBDA (SEL OPERATION)                                  ; Edited 12-Oct-2023 22:36 by rmk
                                                           ; Edited 23-May-2023 12:48 by rmk
                                                           ; Edited 30-May-91 23:00 by jds
    (\DTEST SEL 'SELECTION)
    ;; Set what the selection should be displayed like, given what it's for (NORMAL, COPY, MOVE, etc.)
    (SELECTQ OPERATION
        (NORMAL                                            ; Regular selection
                (FSETSEL SEL HOW BLACKSHADE)
                (FSETSEL SEL HOWHEIGHT 1)
                (FSETSEL SEL HASCARET T))
        (COPY                                              ; Copy source
              (FSETSEL SEL HOW COPYSELSHADE)
              (FSETSEL SEL HOWHEIGHT 1)
              (FSETSEL SEL HASCARET NIL))
        (COPYLOOKS                                         ; copylooks source
                   (FSETSEL SEL HOW COPYLOOKSSELSHADE)
                   (FSETSEL SEL HOWHEIGHT 2)
                   (FSETSEL SEL HASCARET NIL))
        (MOVE                                              ; Copy source
              (FSETSEL SEL HOW EDITMOVESHADE)
              (FSETSEL SEL HOWHEIGHT 16384)
              (FSETSEL SEL HASCARET NIL))
        (DELETE                                            ; To be deleted instantly
                (FSETSEL SEL HOW BLACKSHADE)
                (FSETSEL SEL HOWHEIGHT 16384)
                (FSETSEL SEL HASCARET NIL)
               NIL)
        (PENDINGDEL                                        ; Delete at next type-in
                    (FSETSEL SEL HOW BLACKSHADE)
                    (FSETSEL SEL HOWHEIGHT 16384)
                    (FSETSEL SEL HASCARET T)
                   NIL)
        (INVERTED                                          ; For people who really want to see what's selected.
                  (FSETSEL SEL HOW BLACKSHADE)
                  (FSETSEL SEL HOWHEIGHT 16384)
                  (FSETSEL SEL HASCARET T)
                  (SHOULDNT))
        SEL])

)

(DEFINEQ
```

(\TEDIT.SHOWSEL

```
  [LAMBDA (SEL ON ONLYPANE TEXTOBJ)                        ; Edited 20-Mar-2024 10:56 by rmk
                                                           ; Edited  9-Mar-2024 12:01 by rmk
                                                           ; Edited 18-Feb-2024 15:24 by rmk
                                                           ; Edited 24-Jan-2024 08:07 by rmk
                                                           ; Edited 18-Nov-2023 11:27 by rmk
                                                           ; Edited 14-Oct-2023 12:10 by rmk
                                                           ; Edited  5-Apr-2023 09:13 by rmk
                                                           ; Edited 22-May-92 16:11 by jds
    (\DTEST SEL 'SELECTION)
    ;; Highlight the selection SEL, according to HOW, turning it on or off according to ON. ONLYPANE is specified in calls from \TEDIT.SCROLLFN to
    ;; confine operations to only the pane currently being scrolled.  Other panes are neither unhighlighted or rehighlighted.
    ;; The selection's lines [L1...LN] are the subset of lines selected globally by CH# to CHLIM that are visible within each pane.
    (CL:WHEN (FGETSEL SEL SET)                             ; Nothing to do if not set
        (PROG [(TEXTOBJ (TEXTOBJ! (OR TEXTOBJ (FGETSEL SEL SELTEXTOBJ]
          ;; This operation only makes sense if there is at least one pane to highlight in, and we are allowed to update.
```

```
            (CL:UNLESS (AND (FGETTOBJ TEXTOBJ \WINDOW)
                            (NOT (FGETTOBJ TEXTOBJ TXTDON'TUPDATE)))
                (RETURN))
            (CL:WHEN (EQ ON (FGETSEL SEL ONFLG))                ; No change, nothing to do
                (RETURN))
            (CL:WHEN (FGETSEL SEL SELOBJ))

        ;;

            (if (FGETSEL SEL SELOBJ)
                then ;; SELOBJ if the selection consisted only of a single image object.  It presumably did its own operation when it was
                     ;; selected, but is otherwise immune to normal highlighting.  But it does act just as a normal character in all panes if it is
                     ;; part of a longer selection.

                     (for PANE inpanes (PROGN TEXTOBJ) as L1 in (FGETSEL SEL L1)
                        when (AND L1 (OR (NULL ONLYPANE)
                                         (EQ PANE ONLYPANE)))
                        do (\TEDIT.OBJECT.SHOWSEL TEXTOBJ SEL L1 ON PANE))
                  else (for PANE inpanes (PROGN TEXTOBJ) as L1 in (FGETSEL SEL L1) as LN
                          in (FGETSEL SEL LN) as CARET in (FGETTOBJ TEXTOBJ CARET)
                          when (OR (NULL ONLYPANE)
                                   (EQ PANE ONLYPANE))
                          do (CL:WHEN (AND L1 LN (NEQ 0 (FGETSEL SEL DCH)))
                                                                ; Hilight if not a point selection
                                   (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ L1 LN PANE SEL))
                             (\TEDIT.SETCARET SEL PANE TEXTOBJ ON CARET)))
            (FSETSEL SEL ONFLG ON)))])
```

(\TEDIT.SHOWSEL.HILIGHT

```
  [LAMBDA (TEXTOBJ L1 LN PANE SEL X0 XLIM)                      ; Edited 22-Dec-2023 08:42 by rmk
                                                                ; Edited 17-Dec-2023 17:44 by rmk
                                                                ; Edited 22-Apr-2023 15:32 by rmk
                                                                ; Edited 30-May-91 23:07 by jds
    ;;
    ;; Do the actual highlighting and unhighlighting of a selection for \SHOWSEL.  L1 is the first selected line to be highlighted in PANE, LN is the last
    ;; selected line.  There may be other selected lines visible in other panes but not here.  X0 and XLIM are the x values to be use for the first and last
    ;; lines of the selection, at the ends of the selection within those lines.  LX1 and LXLIM are used for intermediate lines
    (\DTEST L1 'LINEDESCRIPTOR)
    (\DTEST LN 'LINEDESCRIPTOR)
    ;; If the first visible line (L1) is also the first line of the selection, then X0 is the left boundary of the highlight.  Otherwise, the left boundary is the left
    ;; boundary of L1 (its LX1).  The test is (EQ L L1).
    ;;
    ;; Similarly, if the last visible line (LN) is also the last line of the selection, in which case the last boundary of the highlight is XLIM. Otherwise it is
    ;; LN's LXLIM.
    (CL:UNLESS X0
        (SETQ X0 (CL:IF (WITHINLINEP (FGETSEL SEL CH#)
                                     L1)
                        (FGETSEL SEL X0)
                        (FGETLD L1 LX1))))
    (CL:UNLESS XLIM
        (SETQ XLIM (CL:IF (WITHINLINEP (SUB1 (FGETSEL SEL CHLIM))
                                       LN)
                          (FGETSEL SEL XLIM)
                          (FGETLD LN LXLIM))))
    (for L LEFT RIGHT (SHADE _ (OR (FGETSEL SEL HOW)
                                   BLACKSHADE))
        (SHADEHEIGHT _ (OR (FGETSEL SEL HOWHEIGHT)
                           1))
        (PBOTTOM _ (fetch BOTTOM of (DSPCLIPPINGREGION NIL PANE)))
        DISTBELOW first ;; DISTBELOW=1 gives a 1-pt spacing between the line-bottom and the selection underline.  If 0, the slection line runs
                        ;; through the bottom; it makes 1-point horizontal rules invisible.  However:  1 has to be coordinate with
                        ;; \TEDIT.SCROLLUP, so that the selection on the bottom line moves up when the line itself is bltted.  I.e., the visible
                        ;; bottom is one point lower than the bottom of the line.
                        (SETQ DISTBELOW 0)
                        (CL:WHEN (AND (EQ SHADE BLACKSHADE)
                                      (FGETTOBJ TEXTOBJ TXTREADONLY))
                                                                ; Make READONLY selections black.
                            (SETQ SHADEHEIGHT 2))
                        inlines L1
      while (IGEQ (FGETLD L YBOT)
                  PBOTTOM)
      do (SETQ LEFT (OR (AND (EQ L L1)
                             X0)
                        (FGETLD L LX1)))
         (SETQ RIGHT (OR (AND (EQ L LN)
                              XLIM)
                         (FGETLD L LXLIM)))
         (BLTSHADE SHADE PANE LEFT (IDIFFERENCE (FGETLD L YBOT)
                                                DISTBELOW)
                (IDIFFERENCE RIGHT LEFT)
                (IMIN SHADEHEIGHT (FGETLD L LHEIGHT))
                'INVERT)
      repeatuntil (EQ L LN)])
```

(\TEDIT.UPDATE.SHOWSEL

```
  [LAMBDA (NSEL OSEL TEXTOBJ)                                    ; Edited 15-Mar-2024 13:37 by rmk
                                                                 ; Edited 18-Feb-2024 15:25 by rmk
                                                                 ; Edited 17-Nov-2023 23:03 by rmk
                                                                 ; Edited 23-Oct-2023 23:18 by rmk
                                                                 ; Edited  9-Oct-2023 23:02 by rmk
                                                                 ; Edited 14-Jun-2023 16:35 by rmk
                                                                 ; Edited 20-Apr-2023 12:49 by rmk
                                                                 ; Edited  8-Apr-2023 22:49 by rmk
                                                                 ; Edited 21-Oct-2022 18:41 by rmk
                                                                 ; Edited 30-May-91 23:03 by jds
    ;; Update the selection highlighting and caret flashing to represent NSEL. Instead of normal \TEDIT.SHOWSEL, the goal is to avoid changing the
    ;; highlighting on the screen for the pixels that are common between the 2 selections, to just flip whatever bits need to be flipped in order for the
    ;; screen to reflect NSEL's highlighting.  We know that both selections are from the same TEXTOBJ, and that they both implement the same
    ;; operation (copy, delete) so have the same HOW and HOWHEIGHT.  And we know that OSEL is ON.

    ;; It is also the case that the lines of NSEL and OSELfor each pane are drawn from the same lists.

    ;;

    ;; There are 4 cases to consider:

    ;;   1.  The NSEL operates on the first line of OSEL (NSEL POINT)=LEFT

    ;;       a.  (NSEL CH#) < (OSEL CH#).  The old selection is growing to the left--more highlighting

    ;;       b.  (NSEL CH#) > (OSEL CH#).  The old selection is shrinking--less highlighting

    ;;   2.  The NSEL operates on the last line of OSEL:  (NSEL POINT)=RIGHT

    ;;       a.  (NSEL CHLIM) < (OSEL CHLIM).  The old selection is shrinking at the bottom

    ;;       b.  (NSELC CHLIM) > OSEL CHLIM). The old selection is growing.

    ;;

    ;; The nicest implementation would be to create a selection that describes just the characters whose highlighting has changed (either on or off), and
    ;; then do (\TEDIT.SHOWSEL of that selection)

    (LET ((NPOINT (FGETSEL NSEL POINT))
          NEWLINES OLDLINES NCH# OCH# NLASTCH OLASTCH CARETXPOS)
         (FSETSEL NSEL ONFLG T)
         ;;
         (SELECTQ NPOINT
              (LEFT                                              ; Case 1: NSEL changes OSEL top
                    (SETQ NEWLINES (FGETSEL NSEL L1))
                    (SETQ OLDLINES (FGETSEL OSEL L1))
                    (SETQ NCH# (FGETSEL NSEL CH#))
                    (SETQ OCH# (FGETSEL OSEL CH#)))
              (RIGHT (SETQ NEWLINES (FGETSEL NSEL LN))           ; Case 2: NSEL changes OSEL bottom
                     (SETQ OLDLINES (FGETSEL OSEL LN))
                     (SETQ NLASTCH (SUB1 (FGETSEL NSEL CHLIM)))
                     (SETQ OLASTCH (SUB1 (FGETSEL OSEL CHLIM))))
              (SHOULDNT))
         (CL:WHEN (FGETSEL OSEL HASCARET)                        ; Take up the caret in every pane
             (for PCARET in (FGETTOBJ TEXTOBJ CARET) do (\TEDIT.UPCARET PCARET)))
         (for NEWL in NEWLINES as OLDL in OLDLINES as PANE inpanes (PROGN TEXTOBJ) as PCARET
            in (FGETTOBJ TEXTOBJ CARET)
            do (CL:WHEN (AND NEWL OLDL)                          ; NSEL is passed just to provide the shade
                    (SELECTQ NPOINT
                         (LEFT [if (ILESSP (FGETSEL NSEL CH#)
                                           (FGETSEL OSEL CH#))
                                   then                          ; 1a:  Selection-front is growing
                                        (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ NEWL OLDL PANE NSEL (CL:WHEN (WITHINLINEP
                                                                                                      NCH# NEWL)
                                                                                                    (FGETSEL NSEL X0
                                                                                                             ))
                                                                (CL:WHEN (WITHINLINEP OCH# OLDL)
                                                                         (FGETSEL OSEL X0)))
                                   else                          ; 1b:  Selection-front is shrinking
                                        (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ OLDL NEWL PANE NSEL (CL:WHEN (WITHINLINEP
                                                                                                      OCH# OLDL)
                                                                                                    (FGETSEL OSEL X0))
                                                                (CL:WHEN (WITHINLINEP NCH# NEWL)
                                                                         (FGETSEL NSEL X0])
                         (RIGHT [if (IGREATERP (FGETSEL NSEL CHLIM)
                                               (FGETSEL OSEL CHLIM))
                                    then                         ; 2a:  Selection-end is growing
                                         (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ OLDL NEWL PANE NSEL
                                                 (CL:WHEN (WITHINLINEP OLASTCH OLDL)
                                                          (FGETSEL OSEL XLIM))
                                                 (CL:WHEN (WITHINLINEP NLASTCH NEWL)
                                                          (FGETSEL NSEL XLIM)))
                                    else                         ; 2b: Selection-end is shrinking
                                         (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ NEWL OLDL PANE NSEL (CL:WHEN (WITHINLINEP
                                                                                                       NLASTCH NEWL)
                                                                                                     (FGETSEL NSEL
                                                                                                              XLIM))
                                                                 (CL:WHEN (WITHINLINEP OLASTCH OLDL)
                                                                          (FGETSEL OSEL XLIM])
```

```
                    NIL))
            (\TEDIT.SETCARET NSEL PANE TEXTOBJ T PCARET])
```

(\TEDIT.REFRESH.SHOWSEL

```
[LAMBDA (TEXTOBJ SOURCESEL OLDSEL OLDOP NEWOP EXTENDFLG)        ; Edited 15-Mar-2024 13:38 by rmk
                                                                ; Edited  9-Mar-2024 12:02 by rmk
                                                                ; Edited 11-Feb-2024 00:06 by rmk
                                                                ; Edited  9-Feb-2024 15:48 by rmk
                                                                ; Edited 28-Jan-2024 23:27 by rmk
                                                                ; Edited  9-Oct-2023 11:48 by rmk
                                                                ; Edited  6-Oct-2023 12:00 by rmk
                                                                ; Edited 14-Jun-2023 16:35 by rmk
                                                                ; Edited 27-May-2023 15:11 by rmk
                                                                ; Edited 18-Apr-2023 23:54 by rmk
                                                                ; Edited  9-Apr-2023 13:24 by rmk
                                                                ; Edited 30-May-91 23:03 by jds
```

;; Update the screen hilighting to account for the changes that have taken place between OLDSEL and SOURCESEL.

```
(COND
   ((AND EXTENDFLG (EQ OLDOP NEWOP)
         (GETSEL OLDSEL ONFLG))
```

;; If we're extending a selection and the looks haven't changed, we can try doing it the fast way, to prevent flicker.

```
    (\TEDIT.UPDATE.SHOWSEL SOURCESEL OLDSEL TEXTOBJ)
    (\TEDIT.COPYSEL SOURCESEL OLDSEL)
    (SETSEL OLDSEL ONFLG T)
    OLDSEL)
   (T                                                           ; Otherwise, we have to turn the old one off, change things, and
                                                                ; turn the new one on.
      (\TEDIT.SHOWSEL OLDSEL NIL NIL TEXTOBJ)
      (SETSEL OLDSEL SET NIL)
      (CL:UNLESS (EQ OLDOP NEWOP)
             (\TEDIT.SET.SEL.LOOKS SOURCESEL NEWOP))
      (\TEDIT.COPYSEL SOURCESEL OLDSEL)
      (SETSEL OLDSEL ONFLG NIL)                                 ; Make sure we can turn the highlighting on.
      (\TEDIT.SHOWSEL OLDSEL T NIL TEXTOBJ)
      OLDSEL])
```

(\TEDIT.UPDATE.SEL

```
[LAMBDA (SEL CH# DCH POINT DONTFIX)                             ; Edited 15-Mar-2024 13:36 by rmk
                                                                ; Edited  5-Mar-2024 14:45 by rmk
                                                                ; Edited 25-Feb-2024 17:30 by rmk
                                                                ; Edited 16-Feb-2024 23:49 by rmk
                                                                ; Edited 17-Sep-2023 00:05 by rmk
                                                                ; Edited 12-Aug-2023 08:27 by rmk
                                                                ; Edited  6-Jun-2023 13:24 by rmk
                                                                ; Edited  7-May-2023 19:03 by rmk
```

;; Translates the selection SEL to new positions. DCH=0 means point selection with caret blinking either before or after CH#, depending on ;; POINT. If CH# is a history event, that defines the new selection parameters. Otherwise, if any of the variables are NIL, the value for that field in ;; SEL is not changed.

;; Unless DONTFIX, \FIXSEL is called to figure out the pane-lines and screen coordinates.

```
[if (type? TEDITHISTORYEVENT CH#)
    then                                                        ; History is a pseudo-selection
         (CL:UNLESS DCH
             (SETQ DCH (GETTH CH# THLEN)))
         (CL:UNLESS POINT
             (SETQ POINT (GETTH CH# THPOINT CH#)))
         (SETQ CH# (GETTH CH# THCH#))
  else ;; Get defaults from SEL (either a selection or a textobj whose SEL is indicated)
         (CL:WHEN (type? TEXTOBJ SEL)
             (SETQ SEL (TEXTSEL SEL)))
         (CL:UNLESS CH#
             (SETQ CH# (GETSEL SEL CH#)))
         (CL:UNLESS DCH
             (SETQ DCH (FGETSEL SEL DCH)))
         (CL:UNLESS POINT
             (SETQ POINT (FGETSEL SEL POINT)))]
```

;; Restrict CH# to [1..TEXTLEN], using POINT to designate below or above

```
(LET ((TEXTLEN (TEXTLEN (GETSEL SEL SELTEXTOBJ)))
      CHLIM)
     (CL:WHEN (ILESSP CH# 1)
         (SETQ CH# 1)
         (SETQ POINT 'LEFT))
     (CL:WHEN (IGREATERP CH# TEXTLEN)
         (SETQ CH# (ADD1 TEXTLEN))
         (SETQ POINT 'LEFT))
```

;; POINT=LEFT means before CH#, POINT=RIGHT means before CHLIM. If DCH=0, caret is between (and CHLIM - CH# is not DCH=0).

```
     [SETQ CHLIM (CL:IF (EQ DCH 0)
                    (ADD1 CH#)
                    (IMIN (IPLUS CH# DCH)
                          (ADD1 TEXTLEN)))]
```

```
        (SETSEL SEL CH# CH#)
        (FSETSEL SEL DCH DCH)
        (FSETSEL SEL CHLIM CHLIM)
        (FSETSEL SEL POINT POINT)
        (FSETSEL SEL SET T)                                    ;
        (FSETSEL SEL SELOBJ NIL)                               ; If we are moving around, we are moving away from any old
                                                               ; object
        (CL:UNLESS DONTFIX (\TEDIT.FIXSEL SEL))
        SEL])
```

(\TEDIT.SEL.L1

```
  [LAMBDA (SEL PANE TEXTOBJ)                                   ; Edited 16-Nov-2023 23:43 by rmk
    ;; Returns L1 for PANE in SEL
    (for P inpanes (PROGN TEXTOBJ) as L in (GETSEL SEL L1) when (EQ P PANE) do (RETURN L])
```

(\TEDIT.SEL.LN

```
  [LAMBDA (SEL PANE TEXTOBJ)                                   ; Edited 16-Nov-2023 23:43 by rmk
    ;; Returns LN for PANE in SEL
    (for P inpanes (PROGN TEXTOBJ) as L in (GETSEL SEL LN) when (EQ P PANE) do (RETURN L])
```

(\TEDIT.SEL.DELETEDCHARS

```
  [LAMBDA (SELTOFIX TARGETSEL)                                 ; Edited 20-Feb-2024 17:31 by rmk
                                                               ; Edited 15-Feb-2024 23:39 by rmk
                                                               ; Edited 14-Feb-2024 20:59 by rmk
    ;; Adjust SELTOFIX to reflect character number translations after NCHARSDELETED characters starting at FIRSTDELETEDCHAR have been (or
    ;; would be) removed.
    (LET ((FIRSTDELETEDCHNO (FGETSEL TARGETSEL CH#))
          (LASTDELETEDCHNO (SUB1 (FGETSEL TARGETSEL CHLIM)))
          (NCHARSDELETED (FGETSEL TARGETSEL DCH)))
         (CL:WHEN (AND (FGETSEL SELTOFIX SET)
                       (IGEQ (FGETSEL SELTOFIX CH#)
                             FIRSTDELETEDCHNO))
             ;; Nothing to do if SELTOFIX is not set or the deletion happened after the selection.
             [if (ILESSP LASTDELETEDCHNO (FGETSEL SELTOFIX CH#))
                 then ;; All deleted characters are in front of SELTOFIX, just move SETOFIXL forward
                      (add (FGETSEL SELTOFIX CH#)
                           (IMINUS NCHARSDELETED))
                      (add (FGETSEL SELTOFIX CHLIM)
                           (IMINUS NCHARSDELETED))
               else ;; SELTOFIX starts after the last pre-deletion character and is shortened so that it only covers its still-remaining characters.
                    ;; Because of IMAX, this reduces to a point selection if all of SELTOFIX's characters (and more) have been deleted.
                    (\TEDIT.UPDATE.SEL SELTOFIX FIRSTDELETEDCHNO (IMAX 0 (IDIFFERENCE LASTDELETEDCHNO
                                                                                      (SUB1 (FGETSEL SELTOFIX CHLIM])])

)

(DEFINEQ
```

(\TEDIT.COPYSEL

```
  [LAMBDA (FROM TO)                                            ; Edited 24-Jan-2024 09:37 by rmk
                                                               ; Edited 25-Oct-2023 22:24 by rmk
                                                               ; Edited 22-Oct-2023 23:05 by rmk
                                                               ; Edited 23-Apr-2023 12:16 by rmk
                                                               ; Edited  2-Mar-2023 14:55 by rmk
                                                               ; Edited 21-Oct-2022 18:42 by rmk
    (\DTEST FROM 'SELECTION)
    (if TO
        then (\DTEST TO 'SELECTION)
             (FSETSEL TO X0 (FGETSEL FROM X0))
             (FSETSEL TO CH# (FGETSEL FROM CH#))
             (FSETSEL TO XLIM (FGETSEL FROM XLIM))
             (FSETSEL TO CHLIM (FGETSEL FROM CHLIM))
             (FSETSEL TO DCH (FGETSEL FROM DCH))
             (FSETSEL TO L1 (COPY (FGETSEL FROM L1)))
             (FSETSEL TO LN (COPY (FGETSEL FROM LN)))
             (FSETSEL TO POINT (FGETSEL FROM POINT))
             (FSETSEL TO SET (FGETSEL FROM SET))
             (FSETSEL TO SELTEXTOBJ (FGETSEL FROM SELTEXTOBJ))
             (FSETSEL TO SELKIND (FGETSEL FROM SELKIND))
             (FSETSEL TO HOW (FGETSEL FROM HOW))
             (FSETSEL TO HOWHEIGHT (FGETSEL FROM HOWHEIGHT))
             (FSETSEL TO HASCARET (FGETSEL FROM HASCARET))
             (FSETSEL TO SELOBJ (FGETSEL FROM SELOBJ))
             (FSETSEL TO ONFLG (FGETSEL FROM ONFLG))
      else (SETQ TO (create SELECTION using FROM)))
    TO])
```

(\TEDIT.SEL.CHANGED?

```
  [LAMBDA (NEWSEL OLDSEL OLDSELOP NEWSELOP)                  ; Edited 13-Jun-2023 21:50 by rmk
                                                             ; Edited 23-May-2023 12:22 by rmk
                                                             ; Edited  9-Apr-2023 23:15 by rmk
                                                             ; Edited 30-May-91 23:01 by jds
```

;; Decide whether there has been an interesting change in the selection, so we can decide whether to refresh its hilighting on the screen.

```
    (AND NEWSEL (GETSEL NEWSEL SET)
         (NOT (AND (GETSEL OLDSEL SET)
                   (IEQP (GETSEL NEWSEL CH#)
                         (GETSEL OLDSEL CH#))
                   (IEQP (GETSEL NEWSEL CHLIM)
                         (GETSEL OLDSEL CHLIM))
                   (IEQP (GETSEL NEWSEL DCH)
                         (GETSEL OLDSEL DCH))
                   (EQ (GETSEL NEWSEL SELTEXTOBJ)
                       (GETSEL OLDSEL SELTEXTOBJ))
                   (EQ (GETSEL NEWSEL POINT)
                       (GETSEL OLDSEL POINT))
                   (EQ (GETSEL NEWSEL HOW)
                       (GETSEL OLDSEL HOW))
                   (EQ (GETSEL NEWSEL HOWHEIGHT)
                       (GETSEL OLDSEL HOWHEIGHT))
                   (EQ OLDSELOP NEWSELOP])

)

;; SELPIECES

(DEFINEQ
```

(\TEDIT.SELPIECES

```
  [LAMBDA (SEL/FIRSTCHAR LASTCHAR TEXTOBJ)                   ; Edited 17-Mar-2024 00:24 by rmk
                                                             ; Edited  4-Mar-2024 22:47 by rmk
                                                             ; Edited 12-Dec-2023 12:06 by rmk
                                                             ; Edited 11-Dec-2023 10:05 by rmk
                                                             ; Edited  2-Jun-2023 20:36 by rmk
                                                             ; Edited 31-May-2023 10:27 by rmk
                                                             ; Edited  5-Sep-2022 14:40 by rmk
```

;; This converts a selection to the SELPIECES of the properly aligned pieces that SEL/FIRSTCHAR selects. .

;; The first character of SPFIRST is the first character selected in TEXTOBJ and the last character of SPLAST is the last character of the last ;; selected piece in TEXTOBJ. The pieces maintain their chain-sequence pointers in TEXTOBJ. The pieces must be copied and re-chained if they ;; are going to be used in any way that is inconsistent with where they may still be linked into the text.

;;

;; A prefix of the piece containing FIRSTCHAR in TEXTOBJ may be split off, to provide a properly aligned suffix.

;; LIkewise, a suffix of the piece containing LASTCHAR may be split off, to povide a properly aligned prefix.

;; SPLEN is the sum of the lengths of the selected pieces.

;; The I.S.OPR inselpieces iterates over the pieces in SELPIECES.

;;

;; For convenience the "selection" can be specified by FIRSTCHAR and LASTCHAR parameters, plus TEXTOBJ.

```
    (LET (FIRSTCHAR LEFTPC RIGHTPC)
         (if (type? SELECTION SEL/FIRSTCHAR)
             then (SETQ TEXTOBJ (FGETSEL SEL/FIRSTCHAR SELTEXTOBJ))
                  (SETQ FIRSTCHAR (FGETSEL SEL/FIRSTCHAR CH#))
                  [SETQ LASTCHAR (CL:IF (EQ 0 (FGETSEL SEL/FIRSTCHAR DCH))
                                        FIRSTCHAR
                                        (SUB1 (FGETSEL SEL/FIRSTCHAR CHLIM)))]
           elseif (type? TEDITHISTORYEVENT SEL/FIRSTCHAR)
             then (SETQ FIRSTCHAR (GETTH SEL/FIRSTCHAR THCH#))
                  (SETQ LASTCHAR (SUB1 (GETTH SEL/FIRSTCHAR THCHLIM)))
           else (SETQ FIRSTCHAR SEL/FIRSTCHAR))
        ;; Do the right first so that we retain the center piece when FIRTCHAR and LASTCHAR are in the same original piece.
         (SETQ RIGHTPC (\TEDIT.ALIGNEDPIECE (ADD1 LASTCHAR)
                                TEXTOBJ))
         (SETQ LEFTPC (\TEDIT.ALIGNEDPIECE FIRSTCHAR TEXTOBJ))
         (create SELPIECES
                 SPFIRST _ LEFTPC
                 SPLAST _ (PREVPIECE RIGHTPC)
                 SPLEN _ (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR))
                 SPFIRSTCHAR _ FIRSTCHAR
                 SPLASTCHAR _ LASTCHAR])
```

(\TEDIT.SELPIECES.COPY

```
  [LAMBDA (SELPIECES OPERATION TOTEXTOBJ FROMTEXTOBJ)        ; Edited 11-Dec-2023 08:16 by rmk
                                                             ; Edited  2-Jun-2023 11:21 by rmk
                                                             ; Edited 26-May-2023 00:28 by rmk
                                                             ; Edited 21-May-2023 23:01 by rmk
                                                             ; Edited  7-May-2023 17:26 by rmk
```

;; Produces a copy of SELPIECES where the pieces from first to last are chained-together copies of the original pieces so that later inpieces can ;; run from first to last. OPERATION determines which imageobject functions will be invoked, if any.

```
;; FROMTEXTOBJ is optional.  Providing a FROMTEXTOBJ that is different from TOTEXTOBJ is a signal that this is a cross-copy needing to
;; create private copies of strings and files.

(CL:UNLESS FROMTEXTOBJ (SETQ FROMTEXTOBJ TOTEXTOBJ))
(for PC NPC PREVPC NEWFIRSTPIECE inselpieces SELPIECES do (SETQ NPC (\TEDIT.COPYPIECE PC FROMTEXTOBJ
                                                                    TOTEXTOBJ NIL OPERATION))
                                                          (CL:UNLESS NPC
                                                              ; Was an object-copy disallowed?
                                                              (RETURN))
                                                          ;; Linke the new pieces together
                                                          (if PREVPC
                                                              then (replace (PIECE NEXTPIECE) of PREVPC
                                                                      with NPC)
                                                            else (SETQ NEWFIRSTPIECE NPC))
                                                          (replace (PIECE PREVPIECE) of NPC with PREVPC)
                                                          (SETQ PREVPC NPC)
   finally (RETURN (create SELPIECES using SELPIECES SPFIRST _ NEWFIRSTPIECE SPLAST _ PREVPC])
```

(\TEDIT.SELPIECES.CONCAT

```
[LAMBDA (SP1 SP2 TEXTOBJ)                                  ; Edited  3-Mar-2024 12:24 by rmk
                                                           ; Edited 11-Dec-2023 23:03 by rmk
                                                           ; Edited  3-Jun-2023 17:08 by rmk
                                                           ; Edited  2-Jun-2023 12:09 by rmk
                                                           ; Edited 21-May-2023 22:20 by rmk
;; The returned SELPIECE concatenates the pieces in SP1 and SP2.  Probably only sensible if those pieces are consecutive with respect to some
;; textobj or some operation.

;; NOTE:  This modifies the actual pieces to connect them together.  Caller is responsible for insuring that this is safe.

(if (NULL (fetch (SELPIECES SPFIRST) of SP1))
    then SP2
  elseif (NULL (fetch (SELPIECES SPFIRST) of SP2))
    then SP1
  else (freplace (PIECE NEXTPIECE) of (ffetch (SELPIECES SPLAST) of SP1) with (ffetch (SELPIECES SPFIRST)
                                                                                 of SP2))
       (freplace (PIECE PREVPIECE) of (ffetch (SELPIECES SPFIRST) of SP2) with (ffetch (SELPIECES SPLAST)
                                                                                  of SP1))
       (create SELPIECES
               SPFIRST _ (ffetch (SELPIECES SPFIRST) of SP1)
               SPLAST _ (ffetch (SELPIECES SPLAST) of SP2)
               SPLEN _ (IPLUS (ffetch (SELPIECES SPLEN) of SP1)
                              (ffetch (SELPIECES SPLEN) of SP2))
               SPFIRSTCHAR _ (ffetch (SELPIECES SPFIRSTCHAR) of SP1)
               SPLASTCHAR _ (ffetch (SELPIECES SPLASTCHAR) of SP2])
```

(\TEDIT.SELPIECES.CHARTRANSFORM

```
[LAMBDA (SELPIECES CHARFN OBJECTSTOO TEXTOBJ)               ; Edited  3-Mar-2024 12:28 by rmk
                                                           ; Edited 24-May-2023 13:04 by rmk
;; This transforms the characters in SELPIECES according to CHARFN, skipping image objects unless OBJECTSTOO.  The purpose is to allow for
;; character transformations (e.g. case switching) without depending on strings (TEDIT.SELAS.STRING) and character insertion (\INSERTCH) as
;; intermediaries.  Strings can't hold image objects.

;; This smashes the pieces, use crosscopy \SELPIECES.COPY first to protect the document pieces.

(for PC PCONTENTS inselpieces SELPIECES
   do (SETQ PCONTENTS (PCONTENTS PC))
      (SELECTC (PTYPE PC)
           (STRING.PTYPES
                (for I CH (STR _ PCONTENTS) from 1 while (SETQ CH (NTHCHARCODE STR I))
                   do (RPLCHARCODE STR I (APPLY* CHARFN CH TEXTOBJ))))
           (FILE.PTYPES (SETFILEPTR PCONTENTS (PFPOS PC))
                        [if (AND NIL (\IOMODEP PCONTENTS 'BOTH T))
                            then ;; Not clear whether \TEDIT.COPYPIECeS has set things up to allow us to actually smash the
                                 ;; underlying stream. So for now, copy into string space.
                                 (for I from 1 to (PLEN PC) do (\OUTCHAR PCONTENTS (APPLY* CHARFN
                                                                                           (\PEEKCCODE
                                                                                                  PCONTENTS T)
                                                                                           TEXTOBJ)))
                          else ;; This assumes that no file piece has a PLEN greater than \MaxArrayLen characters. We rely on the
                               ;; piece-table reader and writer to guarantee this.  If not, ALLOCSTRING will cause an error.
                               (LET ((FATP (NEQ THINFILE.PTYPE (PTYPE PC)))
                                     STR)
                                    (SETQ STR (ALLOCSTRING (PLEN PC)
                                                           NIL NIL FATP))
                                    (for I from 1 to (PLEN PC) do (RPLCHARCODE STR I (APPLY* CHARFN
                                                                                             (\INCCODE PCONTENTS
                                                                                                    )
                                                                                             TEXTOBJ)))
                                    (FSETPC PC PCONTENTS STR)
                                    (FSETPC PC PTYPE (CL:IF FATP
                                                            FATSTRING.PTYPE
                                                            THINSTRING.PTYPE)])
           (OBJECT.PTYPE (CL:WHEN OBJECTSTOO
```

```
                                (FSETPC PC PCONTENTS (APPLY* CHARFN PCONTENTS TEXTOBJ))))
              (SUBSTREAM.PTYPE
                  (HELP "SUBSTREAM PIECE ?"))
              (SHOULDNT)))
    SELPIECES])
```

(**TEDIT.SELPIECES.FROM.STRING**

```
  [LAMBDA (STRING TEXTOBJ CHECKFOREOL CHARLOOKS PARALOOKS)       ; Edited 20-Mar-2024 10:57 by rmk
                                                                 ; Edited  3-Mar-2024 13:00 by rmk
                                                                 ; Edited 28-Jan-2024 08:28 by rmk
                                                                 ; Edited 11-Dec-2023 08:12 by rmk
                                                                 ; Edited 25-Nov-2023 15:22 by rmk
                                                                 ; Edited 11-Nov-2023 15:49 by rmk
                                                                 ; Edited  2-Jun-2023 11:59 by rmk
                                                                 ; Edited 24-May-2023 15:26 by rmk
    ;; Creates SELPIECES with pieces representing STRING.  If CHECKFOREOL and the string contains a paragraph-breaking character, then the
    ;; string will be coded as a sequence of pieces with EOL-terminated pieces (but not necessarily the last piece) marked as PPARALAST.
    (TEXTOBJ! TEXTOBJ)
    (CL:UNLESS CHARLOOKS
        (SETQ CHARLOOKS (FGETTOBJ TEXTOBJ DEFAULTCHARLOOKS)))
    (CL:UNLESS PARALOOKS
        (SETQ PARALOOKS (FGETTOBJ TEXTOBJ FMTSPEC)))
    (CL:WHEN (AND TEXTOBJ (FGETTOBJ TEXTOBJ FORMATTEDP))
           (SETQ CHECKFOREOL T))
    (LET (FIRSTPIECE EOLPOS (BYTESPERCHAR 1)
                (PTYPE THINSTRING.PTYPE)
                (PBINABLE T))
         (SETQ STRING (CONCAT STRING))
         (CL:WHEN (fetch (STRINGP FATSTRINGP) of STRING)
             (SETQ PTYPE FATSTRING.PTYPE)
             (SETQ PBINABLE NIL)
             (SETQ BYTESPERCHAR 2))
         (if (AND CHECKFOREOL (SETQ EOLPOS (STRPOS (CONSTANT (CHARACTER (CHARCODE EOL)))
                                                   STRING)))
             then ;; Break it up into PPARALAST pieces
                  [bind PC STR PREVPC (NCHARS _ (NCHARS STRING))
                        (LASTEOLPOS _ 0)
                     collect (SETQ STR (SUBSTRING STRING (ADD1 LASTEOLPOS)
                                                  (SETQ LASTEOLPOS EOLPOS)))
                             (PROG1
                                 (SETQ PC
                                  (create PIECE
                                         PTYPE _ PTYPE
                                         PCONTENTS _ STR
                                         PLEN _ (NCHARS STR)
                                         PBYTELEN _ (ITIMES (NCHARS STR)
                                                            BYTESPERCHAR)
                                         PLOOKS _ CHARLOOKS
                                         PPARALOOKS _ PARALOOKS
                                         PPARALAST _ T
                                         PREVPIECE _ PC
                                         PBINABLE _ PBINABLE))
                                 (CL:WHEN PREVPC (FSETPC PREVPC NEXTPIECE PC))
                                 (SETQ PREVPC PC)
                                 (SETQ EOLPOS (OR (STRPOS (CONSTANT (CHARACTER (CHARCODE EOL)))
                                                          STRING
                                                          (ADD1 LASTEOLPOS))
                                                  NCHARS)))
                     repeatuntil (IGEQ LASTEOLPOS NCHARS)
                     finally (CL:UNLESS (EQ (CHARCODE EOL)
                                            (NTHCHARCODE STR -1))   ; Last piece didn't end in EOL
                                 (FSETPC PC PPARALAST NIL))
                             (RETURN (create SELPIECES
                                             SPFIRST _ (CAR $$VAL)
                                             SPLAST _ PC
                                             SPLEN _ NCHARS
                                             SPFIRSTCHAR _ 1
                                             SPLASTCHAR _ (NCHARS STRING]
           else (SETQ FIRSTPIECE (create PIECE
                                         PTYPE _ PTYPE
                                         PCONTENTS _ STRING
                                         PLEN _ (NCHARS STRING)
                                         PBYTELEN _ (ITIMES (NCHARS STRING)
                                                            BYTESPERCHAR)
                                         PBYTESPERCHAR _ BYTESPERCHAR
                                         PBINABLE _ PBINABLE
                                         PLOOKS _ CHARLOOKS
                                         PPARALOOKS _ PARALOOKS))
                (create SELPIECES
                        SPFIRST _ FIRSTPIECE
                        SPLAST _ FIRSTPIECE
                        SPLEN _ (NCHARS STRING)
                        SPFIRSTCHAR _ 1
                        SPLASTCHAR _ (NCHARS STRING])
```

(\TEDIT.SELPIECES.TO.STRING

```
  [LAMBDA (SELPIECES OBJECTCHARCODE TEXTOBJ)                    ; Edited  3-Mar-2024 12:24 by rmk
                                                                ; Edited  2-Jun-2023 12:07 by rmk
                                                                ; Edited 24-May-2023 20:00 by rmk
    ;; Produce a string representing the contents of SELPIECES.  Optional OBJECTCHARCODE is a code to be used to represent an image object.  If
    ;; it is a TEXTOBJ with an OBJECTBYTE property, then that code is used.  Otherwise, arbitrarily the escape character.

    ;; Would it be better to take the chracters from the PREPRINTFN, if present?

    (for PC PCONTENTS (I _ 1)
        (RESULT _ (ALLOCSTRING (fetch (SELPIECES SPLEN) of SELPIECES)))
        inselpieces SELPIECES do (SETQ PCONTENTS (PCONTENTS PC))
                                 (SELECTC (PTYPE PC)
                                      (STRING.PTYPES
                                           (RPLSTRING RESULT I PCONTENTS)
                                           (add I (PLEN PC)))
                                      (FILE.PTYPES (SETFILEPTR PCONTENTS (PFPOS PC))
                                                   (for J from 1 to (PLEN PC) do (RPLCHARCODE RESULT I
                                                                                              (\INCCODE PCONTENTS))
                                                                                 (add I 1)))
                                      (OBJECT.PTYPE             ; Could run the PREPRINTFN ?  But we then have to let the
                                                                ; string grow.
                                                    (RPLCHARCODE RESULT I (OR (SMALLP OBJECTCHARCODE)
                                                                              [AND (SETQ OBJECTCHARCODE
                                                                                         (GETTEXTPROP TEXTOBJ
                                                                                                      'OBJECTBYTE]
                                                                              (CHARCODE ESCAPE)))
                                                    (add I 1))
                                      (SUBSTREAM.PTYPE
                                          (HELP "SUBSTREAM PIECE?"))
                                      (SHOULDNT))
       finally (RETURN RESULT])

)

;; User entries to the selection code

(DEFINEQ

(TEDIT.XYTOCH
  [LAMBDA (X Y PANE)                                            ; Edited 20-Mar-2024 14:32 by rmk
    ;; Returns the character number of the character at coordinates X and Y in PANE.
    (LET ((TEXTOBJ (TEXTOBJ PANE))
          SEL)
         ;; The X W fields should be good in all panes, not sure about the Y W fields.  Maybe those are PANE-dependent.
         (SETQ X (SELECTQ X
                      (LEFT (GETTOBJ TEXTOBJ WLEFT))
                      (RIGHT (SUB1 (GETTOBJ TEXTOBJ WRIGHT)))
                      X))
         (SETQ Y (SELECTQ Y
                      (TOP (SUB1 (GETTOBJ TEXTOBJ WTOP)))
                      (BOTTOM (GETTOBJ TEXTOBJ WBOTTOM))
                      Y))
         (SETQ SEL (\TEDIT.SELECT.LINE.SCANNER X Y TEXTOBJ (fetch (TEXTWINDOW PLINES) of PANE)
                           'TEXT NIL NIL PANE))
         (CL:WHEN (AND (type? SELECTION SEL)
                       (GETSEL SEL SET))                        ; He pointed at something real;  return that.
             (GETSEL SEL CH#))])

(TEDIT.GETPOINT
  [LAMBDA (STREAM SEL)                                          ; Edited  5-Jun-2023 15:30 by rmk
                                                                ; Edited 30-May-91 23:03 by jds
    ;; Given a selection, tell the CHNO that type-in would be inserted in front of.  IF SEL is given, use it to decide.  Otherwise, use STREAM's current
    ;; selection.  SEL can also be a character number, which is simply returned.

    (CL:UNLESS SEL
        (SETQ SEL (TEXTSEL (TEXTOBJ STREAM))))
    (if (NOT (type? SELECTION SEL))
        then SEL
      elseif (FGETSEL SEL SET)
        then ;; LEFT and RIGHT are the same for a point (DCH=0) selection.
             (SELECTQ (FGETSEL SEL POINT)
                  (LEFT (FGETSEL SEL CH#))
                  (RIGHT (FGETSEL SEL CHLIM))
                  (SHOULDNT "Selection's POINT is neither RIGHT nor LEFT."])

(TEDIT.GETSEL
  [LAMBDA (TSTREAM)                                             ; Edited  1-May-2023 21:07 by rmk
                                                                ; Edited 30-May-91 23:03 by jds
    (create SELECTION using (fetch (TEXTOBJ SEL) of (TEXTOBJ TSTREAM])
```

(TEDIT.GETSEL.PARA

```
[LAMBDA (TSTREAM)                                                    ; Edited 16-Jan-2024 14:59 by rmk
                                                                     ; Edited  1-May-2023 21:07 by rmk
                                                                     ; Edited 30-May-91 23:03 by jds
```

;; Returns a selection that runs from the beginning of the paragraph containing the first currently selected character to the end of the paragraph that ;; contains the last currently selected character.

```
(LET* [(TEXTOBJ (TEXTOBJ TSTREAM))
       (SEL (FGETTOBJ TEXTOBJ SEL))
       [PCH# (CAR (\TEDIT.PARA.FIRST TEXTOBJ (GETSEL SEL CH#]
       (PCHLIM (ADD1 (CAR (\TEDIT.PARA.LAST TEXTOBJ (SUB1 (GETSEL SEL CHLIM]
      (create SELECTION using SEL CH# _ PCH# CHLIM _ PCHLIM DCH _ (IDIFFERENCE PCHLIM PCH#)
                              ONFLG _ NIL SET _ T])
```

(TEDIT.MAKESEL

```
[LAMBDA (STREAM CH# LEN POINT)                                       ; Edited 15-Mar-2024 13:36 by rmk
                                                                     ; Edited  9-Mar-2024 12:03 by rmk
                                                                     ; Edited 16-Jan-2024 14:52 by rmk
                                                                     ; Edited 23-May-2023 12:39 by rmk
                                                                     ; Edited 18-Apr-2023 23:53 by rmk
                                                                     ; Edited 21-Oct-2022 18:37 by rmk
                                                                     ; Edited 30-May-91 23:03 by jds
  (LET* ((TEXTOBJ (TEXTOBJ STREAM))
         (SEL (FGETTOBJ TEXTOBJ SEL)))
        (\TEDIT.SHOWSEL SEL NIL NIL TEXTOBJ)
        (FSETSEL SEL CH# CH#)
        (FSETSEL SEL CHLIM (IMAX CH# (IPLUS CH# LEN)))
        (FSETSEL SEL DCH LEN)
        (FSETSEL SEL POINT (OR POINT 'LEFT))
        (FSETSEL SEL SELTEXTOBJ TEXTOBJ)
        (FSETSEL SEL SET T)
        (\TEDIT.FIXSEL SEL TEXTOBJ)
        (\TEDIT.SHOWSEL SEL T NIL TEXTOBJ])
```

(TEDIT.SCANSEL

```
[LAMBDA (TSTREAM)                                                    ; Edited 21-Mar-2024 10:49 by rmk
                                                                     ; Edited 17-Mar-2024 12:07 by rmk
                                                                     ; Edited 26-May-2023 22:35 by rmk
                                                                     ; Edited  8-Sep-2022 23:29 by rmk
                                                                     ; Edited 30-May-91 23:03 by jds
```

;; Set up to read the selected text; return the sel's length or NIL if nothing selected.

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(LET ((SEL (FGETTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)
                     SEL)))
     (CL:WHEN (GETSEL SEL SET)
         (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 (FGETSEL SEL CH#)))
         (FGETSEL SEL DCH))])
```

(TEDIT.SET.SEL.LOOKS

```
[LAMBDA (SEL OPERATION)                                              ; Edited 15-Mar-2024 13:34 by rmk
                                                                     ; Edited  9-Mar-2024 12:04 by rmk
                                                                     ; Edited 12-Oct-2023 22:32 by rmk
                                                                     ; Edited 10-Jun-2023 13:35 by rmk
                                                                     ; Edited 20-May-2023 23:53 by rmk
                                                                     ; Edited 18-Apr-2023 23:53 by rmk
                                                                     ; Edited 30-May-91 23:01 by jds
```

;; Set what the selection should be displayed like, given what it's for (NORMAL, COPY, MOVE, etc.). This is a documented entry.

```
(LET ((WASON (GETSEL SEL ONFLG)))
     (\TEDIT.SHOWSEL SEL NIL NIL (FGETSEL SEL SELTEXTOBJ))
     (\TEDIT.SET.SEL.LOOKS SEL OPERATION)
     (\TEDIT.SHOWSEL SEL WASON NIL (FGETSEL SEL SELTEXTOBJ))
     SEL])
```

(TEDIT.SETSEL

```
[LAMBDA (STREAM CH# LEN POINT PENDINGDELFLG LEAVECARETLOOKS OPERATION)
                                                                     ; Edited 17-Mar-2024 00:27 by rmk
                                                                     ; Edited 15-Mar-2024 13:38 by rmk
                                                                     ; Edited  9-Mar-2024 12:04 by rmk
                                                                     ; Edited 22-Sep-2023 18:09 by rmk
                                                                     ; Edited  3-Aug-2023 23:12 by rmk
                                                                     ; Edited 23-May-2023 16:50 by rmk
                                                                     ; Edited 18-Apr-2023 23:54 by rmk
                                                                     ; Edited 27-Mar-2023 13:07 by rmk
                                                                     ; Edited 30-May-91 23:05 by jds
```

;; Given a text stream or textobj, and a piece of text to select, set the internal selection, and return it.

```
(LET ((TEXTOBJ (TEXTOBJ STREAM))
      SEL TEXTLEN PC)
     (SETQ SEL (TEXTSEL TEXTOBJ))
```

```
        (SETQ TEXTLEN (TEXTLEN TEXTOBJ))
        (\TEDIT.SHOWSEL SEL NIL NIL TEXTOBJ)                    ; First turn the old sel off.
        [COND
           ((type? SELECTION CH#)                               ; He gave use a selection;  just plug it in
            (\TEDIT.COPYSEL CH# SEL)                            ; And make sure it can be turned on.
            (SETSEL SEL ONFLG NIL))
           (T                                                   ; Documentation doesn't allow NIL, but DINFO.SHOWSEL
                                                                ; passes it
              (SELECTQ POINT
                   (LEFT)
                   (RIGHT)
                   (NIL (SETQ POINT 'LEFT))
                   (ERROR POINT "is an illegal POINT"))         ; He fed us numbers;  use them
              (SETQ LEN (IMAX 0 LEN))                           ; Length must be positive
              (SETQ CH# (IMIN (IMAX 1 CH#)
                              (ADD1 TEXTLEN)))                  ; Starting character. If beyond  TEXTLEN, then just after EOF
              (SETSEL SEL CH# CH#)
              [SETSEL SEL CHLIM (IMAX CH# (IMIN (IPLUS CH# LEN)
                                                (ADD1 TEXTLEN]
              ;; LEN may have been reduced by TEXTLEN
              (SETQ LEN (IDIFFERENCE (GETSEL SEL CHLIM)
                                     (GETSEL SEL CH#)))
              (SETSEL SEL DCH LEN)
              (SETSEL SEL POINT (if (IGREATERP CH# TEXTLEN)
                                    then 'LEFT
                                  elseif POINT
                                  else 'LEFT))                  ; Which side the caret should go on
              (FSETSEL SEL SELOBJ (CL:WHEN (EQ 1 LEN)           ; If CH# beyond TEXTLEN, LEN is 0
                                           (SETQ PC (\TEDIT.CHTOPC (GETSEL SEL CH#)
                                                                   TEXTOBJ))
                                           (CL:WHEN (EQ OBJECT.PTYPE (PTYPE PC))
                                                    (PCONTENTS PC)))]
        (SETSEL SEL SELTEXTOBJ TEXTOBJ)                         ; Link it back to the associated textobj
        [COND
           [PENDINGDELFLG                                       ; This selection is to be a pending-deletion sel.
                  (SETTOBJ TEXTOBJ BLUEPENDINGDELETE T)         ; Warn TEdit that there's a deletion pending
                  (\TEDIT.SET.SEL.LOOKS SEL (OR OPERATION 'PENDINGDEL]
           (T                                                   ; This selection is to be a pending-deletion sel.
              (\TEDIT.RESET.EXTEND.PENDING.DELETE SEL TEXTOBJ)
              (\TEDIT.SET.SEL.LOOKS SEL (OR OPERATION 'NORMAL]
        (SETSEL SEL SET T)                                      ; Mark the selection as valid for others to use
        (CL:UNLESS LEAVECARETLOOKS                              ; And set the insertion looks to follow.
            (SETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL)))
        (\TEDIT.FIXSEL SEL TEXTOBJ)                             ; Update the selection's screen location
        (\TEDIT.SHOWSEL SEL T NIL TEXTOBJ)                      ; Highlight it on the screen
        SEL])
```

(TEDIT.SHOWSEL

```
  [LAMBDA (STREAM ONFLG SEL)                                    ; Edited 15-Mar-2024 13:36 by rmk
                                                                ; Edited  9-Mar-2024 12:06 by rmk
                                                                ; Edited  3-May-2023 09:23 by rmk
                                                                ; Edited 18-Apr-2023 23:54 by rmk
                                                                ; Edited 21-Oct-2022 18:36 by rmk
    ;; He's giving us a selection to highlight and to  connect it to this textobj.   ; Edited 30-May-91 23:04 by jds
    (LET ((TEXTOBJ (TEXTOBJ STREAM)))
         (CL:UNLESS SEL
             (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
         (CL:WHEN SEL
             (SETSEL SEL SELTEXTOBJ TEXTOBJ)
             (\TEDIT.FIXSEL SEL TEXTOBJ)
             (\TEDIT.SHOWSEL SEL ONFLG NIL TEXTOBJ))])
```

(TEDIT.SEL.AS.STRING

```
  [LAMBDA (TSTREAM SEL)                                         ; Edited 17-Mar-2024 12:05 by rmk
                                                                ; Edited 27-Jan-2024 22:57 by rmk
                                                                ; Edited 23-May-2023 12:36 by rmk
                                                                ; Edited  8-Sep-2022 23:35 by rmk
                                                                ; Edited 22-Apr-93 16:44 by jds
    ;; RMK:  WHAT IF THE STREAM CONTAINS AN OBJECT?
    ;; Given a text stream, go to the TEXTOBJ, get the current selection, and return it as a string.
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    (CL:UNLESS SEL
        (SETQ SEL (GETTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)
                           SEL)))
    (LET (RESULT (LEN (GETSEL SEL DCH)))
         (COND
            ((ZEROP LEN)                                        ; There is no selection, or it's zero-width.  Return ''
             (CONCAT ""))
            (T (SETQ RESULT (ALLOCSTRING LEN (CHARCODE SPACE)))  ; The resulting string
               (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 (GETSEL SEL CH#)))
                                                                ; Starting point for the string is start of selection.
               (for I from 1 to LEN do (RPLCHARCODE RESULT I (BIN TSTREAM)))
```

```
                RESULT])
```

(TEDIT.SEL.AS.SEXPR

```
  [LAMBDA (TSTREAM SEL RDTBL FLG)                                    ; Edited 17-Mar-2024 12:05 by rmk
                                                                     ; Edited 25-Dec-2023 18:52 by rmk
                                                                     ; Edited  9-Jul-2023 09:37 by rmk
                                                                     ; Edited 22-Apr-93 16:44 by jds
```

;; Return an s-expression from the characters defined by SEL's point position.

;; This backs up to the beginning of the word that precedes the caret, then READ's from there. A little tricky to point to the paren in front of an ;; atom, to get a complete list structure and not just the initial atom.

```
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    [\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 (\TEDIT.WORD.FIRST (TEXTOBJ TSTREAM)
                                                (TEDIT.GETPOINT TSTREAM SEL)
                                                (TEDIT.ATOMBOUND.READTABLE (OR RDTBL *READTABLE*]
    (READ TSTREAM RDTBL FLG])
```

(TEDIT.SELECTALL

```
  [LAMBDA (TEXTSTREAM TEXTOBJ SEL)                                   ; Edited 14-Jun-2023 16:58 by rmk
                                                                     ; Edited  3-May-2020 17:29 by rmk:
    (TEDIT.SETSEL TEXTSTREAM 0 (ADD1 (TEXTLEN (TEXTOBJ TEXTSTREAM)))
          'LEFT])

)
```

FUNCTION INDEX

MACRO INDEX

CONSTANT INDEX

RECORD INDEX

VARIABLE INDEX

I.S.OPR INDEX