File created: 6-Mar-2025 11:59:08 **{WMEDLEY}<library>TEDIT>TEDIT-SELECTION.;661**

edit by: rmk

changes to: (FNS \TEDIT.CHTOLINEX)

previous date: 28-Feb-2025 17:45:33 {WMEDLEY}<library>TEDIT>TEDIT-SELECTION.;660

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-SELECTIONCOMS
       [(DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS SELECTION SELPIECES)
                                                 (CONSTANTS (COPYSELSHADE 30583)
                                                        (COPYLOOKSSELSHADE 30583)
                                                        (EDITMOVESHADE BLACKSHADE)
                                                        (EDITGRAY 32800))
                                                 (MACROS WITHINLINEP FWITHINLINEP LINESELECTEDP FLINESELECTEDP
                                                        IBETWEENP)
                                                 (MACROS GETSEL SETSEL FGETSEL FSETSEL SELECTION!)
                                                 (I.S.OPRS inselpieces)
                                                 (MACROS GETSPC SETSPC FGETSPC FSETSPC SELPIECES!)
                                                 (GLOBALVARS TEDIT.EXTEND.PENDING.DELETE)
                                                 (GLOBALVARS TEDIT.SELECTION TEDIT.SHIFTEDSELECTION
                                                        TEDIT.MOVESELECTION TEDIT.COPYLOOKSSELECTION
                                                        TEDIT.DELETESELECTION)))
        (INITRECORDS SELECTION SELPIECES)
        (FNS \TEDIT.SELECTION.DEFPRINT)
        (P (\TEDIT.SET.GLOBAL.SELECTIONS))
        (FNS \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.SCAN.LINE \TEDIT.SCAN.LINE.WORD \TEDIT.XYTOSEL \TEDIT.REGIONTYPE
                   \TEDIT.XYTOSEL.INLINEP \TEDIT.XYTOSEL.LINE)
              (FNS \TEDIT.FIXSEL \TEDIT.CHTOLINEX)
              (FNS \TEDIT.RESET.EXTEND.PENDING.DELETE \TEDIT.SET.SEL.LOOKS)
              (FNS \TEDIT.SHOWSEL \TEDIT.SHOWSEL.HILIGHT \TEDIT.UPDATE.SEL \TEDIT.CARETLINE \TEDIT.SEL.L1
                   \TEDIT.SEL.LN \TEDIT.SEL.DELETEDCHARS)
              (FNS \TEDIT.COPYSEL \TEDIT.SEL.CHANGED?))
        (COMS                                                   ; Image objects
              (FNS \TEDIT.SELECT.OBJECT \TEDIT.SHOWSEL.OBJECT \TEDIT.CLIP.OBJECT \TEDIT.OPERATE.OBJECT))
        ;; 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.SELPROP TEDIT.GETPOINT TEDIT.GETSEL TEDIT.GETSEL.PARA TEDIT.SCANSEL
             TEDIT.SET.SEL.LOOKS TEDIT.SETSEL TEDIT.SHOWSEL TEDIT.SEL.AS.STRING TEDIT.SEL.AS.SEXPR
             TEDIT.SELECTALL)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                            (NLAML)
                                                                            (LAMA TEDIT.SELPROP])

(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
                     SELLINES                                   ; A list of (L1 L2) pairs one for each pane, to replace the
                                                                ; separate L1 L2 lists. 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
                    (SELTEXTSTREAM FULLXPOINTER)            ; TEXTSTREAM 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 ((SELTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of (GETSEL DATUM SELTEXTSTREAM)))
                   (CHLAST (STANDARD (SUB1 (GETSEL DATUM CHLIM))
                                     (SETSEL DATUM CHLIM (ADD1 NEWVALUE)))
                           (FAST (SUB1 (FSETSEL DATUM CHLIM))
                                 (FSETSEL DATUM CHLIM (ADD1 NEWVALUE]
       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)

(RPAQ EDITMOVESHADE BLACKSHADE)

(RPAQQ EDITGRAY 32800)

(CONSTANTS (COPYSELSHADE 30583)
       (COPYLOOKSSELSHADE 30583)
       (EDITMOVESHADE BLACKSHADE)
       (EDITGRAY 32800))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS WITHINLINEP MACRO (OPENLAMBDA (CHNO LINE)
                              (AND (IGEQ CHNO (GETLD LINE LCHAR1))
                                   (ILESSP CHNO (FGETLD LINE LCHARLIM))
                                   LINE)))

(PUTPROPS FWITHINLINEP MACRO (OPENLAMBDA (CHNO LINE)
                               (AND (IGEQ CHNO (FGETLD LINE LCHAR1))
                                    (ILESSP CHNO (FGETLD LINE LCHARLIM))
                                    LINE)))

(PUTPROPS LINESELECTEDP MACRO (OPENLAMBDA (L CH# CHLAST)
                                (AND (IGEQ (GETLD L LCHARLAST)
                                           CH#)
                                     (ILEQ (FGETLD L LCHAR1)
                                           CHLAST))))

(PUTPROPS FLINESELECTEDP MACRO (OPENLAMBDA (L CH# CHLAST)      ; True if a CH#..CHLAST selection would include L
                                 (AND (IGREATERP (FGETLD L LCHARLIM)
                                                 CH#)
                                      (ILEQ (FGETLD L LCHAR1)
                                            CHLAST))))

(PUTPROPS IBETWEENP MACRO (OPENLAMBDA (X LOW HIGH)
                            (AND (IGEQ X LOW)
                                 (ILEQ X HIGH))))
)

(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)))

(PUTPROPS SELECTION! MACRO ((SEL)
                            (\DTEST SEL 'SELECTION)))
)

(DECLARE%: EVAL@COMPILE

(I.S.OPR 'inselpieces NIL '[SUBST (GETDUMMYVAR)
                                  '$$SELPIECES
                                  '(BIND $$SPFIRST $$SPLAST $$SPLENGTH $$SELPIECES _ BODY
                                      DECLARE (LOCALVARS $$SELPIECES $$SPFIRST $$SPLAST $$SPLENGTH)
                                      FIRST (\DTEST (OR $$SELPIECES (GO $$OUT))
                                                    'SELPIECES)
                                            [SETQ I.V. (SETQ $$SPFIRST (\DTEST (ffetch (SELPIECES SPFIRST)
                                                                                  of $$SELPIECES)
                                                                               'PIECE]
                                            (SETQ $$SPLAST (\DTEST (ffetch (SELPIECES SPLAST) of $$SELPIECES)
                                                                   'PIECE))
                                            (SETQ $$SPLENGTH (ffetch (SELPIECES SPLEN) of $$SELPIECES))
                                      REPEATUNTIL (EQ I.V. $$SPLAST) BY (\DTEST (NEXTPIECE I.V.)
                                                                                'PIECE]
       T)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETSPC MACRO ((SP FIELD)
                        (fetch (SELPIECES FIELD) of SP)))

(PUTPROPS SETSPC MACRO ((SP FIELD NEWVALUE)
                        (replace (SELPIECES FIELD) of SP with NEWVALUE)))

(PUTPROPS FGETSPC MACRO ((SP FIELD)
                         (ffetch (SELPIECES FIELD) of SP)))

(PUTPROPS FSETSPC MACRO ((SP FIELD NEWVALUE)
                         (freplace (SELPIECES FIELD) of SP with NEWVALUE)))

(PUTPROPS SELPIECES! MACRO ((SPC)
                            (\DTEST SPC 'SELPIECES)))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.EXTEND.PENDING.DELETE)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

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

;; 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)                                                    ; Edited 26-Nov-2024 02:59 by rmk
                                                                   ; Edited 23-Aug-2024 00:16 by rmk
                                                                   ; Edited 29-Apr-2024 12:47 by rmk
                                                                   ; 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 (TEXTOBJ SEL T))
          WHICH INFO LOC)
         (SETQ WHICH (CL:IF (AND TEXTOBJ (EQ SEL (TEXTSEL TEXTOBJ)))
                            'SEL
                            ""))
         (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 "{SEL:" WHICH " " INFO " " (CAR LOC)
                       "/"
                       (CDR LOC)
                       "}"])

)

(\TEDIT.SET.GLOBAL.SELECTIONS)

(DEFINEQ
```

(\TEDIT.SET.GLOBAL.SELECTIONS

; Edited 7-Nov-2024 21:50 by rmk
; Edited 4-Oct-2024 08:39 by rmk
; Edited 15-Mar-2024 13:38 by rmk
; Edited 12-Feb-2024 08:15 by rmk

```
  [LAMBDA (SELOPERATION SOURCESEL)
```

;; 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))))
         (\TEDIT.THELP "UNKNOWN SELOPERATION"])

)

(DEFINEQ
```

(\TEDIT.SELECTED.PIECES

; Edited 26-Nov-2024 10:54 by rmk
; 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

```
  [LAMBDA (TEXTOBJ SEL CROSSCOPY PIECEMAPFN FNARG1 FNARG2)
```

;; 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

```
    (NOTUSED)
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (LET ((SELPIECES (\TEDIT.SELPIECES (OR SEL (TEXTSEL TEXTOBJ))
                                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

; Edited 9-Jul-2024 18:19 by rmk
; 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

```
  [LAMBDA (TEXTOBJ CH# LIMITCH#)
```

; Edited 22-Aug-2022 13:21 by rmk
; Edited 18-Apr-93 23:49 by jds

;; Returns the number of the last protected character before CH# but after LIMITCH#, NIL if nothing later.

```
(SETQ LIMITCH# (IMIN LIMITCH# (TEXTLEN TEXTOBJ)))
(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])
```

(\TEDIT.FIND.PROTECTED.START

```
  [LAMBDA (TEXTOBJ CH# LIMITCH#)
```

; Edited 9-Jul-2024 18:17 by rmk
; 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

;; Returns the number of the first protected character after CH# but before LIMITCH#, NIL if nothing earlier.

```
(LET (START-OF-PIECE)                                   ; Gets us to the beginning of CH# piece
     (DECLARE (SPECVARS START-OF-PIECE))
     ;; Move forward thru the pieces of the document, looking for one that contains protected text.
     (for PC inpieces (\TEDIT.CHTOPC CH# TEXTOBJ T) while (ILESSP START-OF-PIECE LIMITCH#)
        do (CL:WHEN (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC))
                ;; We found the beginning of a protected region.
                (RETURN START-OF-PIECE))
            (add START-OF-PIECE (PLEN PC])
```

(\TEDIT.WORD.BOUND

```
  [LAMBDA (TEXTOBJ PREVCH CH)
```

; Edited 16-Jul-2024 19:52 by rmk
; Edited 27-Sep-2022 23:54 by rmk
; Edited 25-Sep-2022 23:48 by rmk
; Edited 30-May-91 23:02 by jds

```
    (if (AND (FIXP PREVCH)
             (FIXP CH))
        then (LET [(READSA (fetch READSA of (OR (fetch (TEXTOBJ TXTWTBL) of TEXTOBJ)
                                                TEDIT.WORDBOUND.READTABLE]
                  (NEQ (\SYNCODE READSA PREVCH)
                       (\SYNCODE READSA CH)))
      else T])

)

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

;; Setting for a "Laurel" mode

;; Selection manipulating code

```
(DEFINEQ
```

(\TEDIT.EXTEND.SEL

```
  [LAMBDA (NEWSEL CURSEL TEXTOBJ EVENIFPROTECTED)
```

; Edited 11-Sep-2024 23:44 by rmk
; Edited 9-Sep-2024 09:28 by rmk
; Edited 28-Aug-2024 09:49 by rmk
; Edited 22-Aug-2024 16:06 by rmk
; Edited 13-Jul-2024 12:48 by rmk
; Edited 29-Apr-2024 13:45 by rmk
; 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 19-Apr-2023 17:36 by rmk
; Edited 19-Apr-93 13:49 by jds

;;

;; At entry the display is compatible with CURSEL. This function modifies CURSEL and the dispaly to reflect how CURSEL is extended by
;; NEWSEL's coordinates. The strategy is to fix NEWSEL so that it describes the highlighting difference between the original and the extension,
;; and to update the display accordingly.

;; If NEWSEL is before or after CURSEL, the modified NEWSEL describes the intervening characters. If NEWSEL overlaps CURSEL, NEWSEL is
;; reduced to describe the prefix or suffix that should no longer be highlighted. ;

;; 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:WHEN (AND (FGETSEL NEWSEL SET)
                  (FGETSEL CURSEL SET))
         (LET ((CCH# (FGETSEL CURSEL CH#))
               (CCHLIM (FGETSEL CURSEL CHLIM))
               (NCH# (FGETSEL NEWSEL CH#))
               (NCHLIM (FGETSEL NEWSEL CHLIM))
               TEMP)
              (CL:UNLESS EVENIFPROTECTED
                  ;; OLDSEL and NEWSEL do not cover any protected characters, but there may be protected characters between them.  That's
                  ;; OK if we are copying, otherwise reduce NEWSEL to a point selection that excludes the protected characters..

                  ;; Protection may only be relevant for menus.

                  (if (IGREATERP NCH# CCHLIM)
                      then ;; New is later
                           (CL:WHEN (SETQ TEMP (\TEDIT.FIND.PROTECTED.START TEXTOBJ CCHLIM NCH#))
                               (\TEDIT.UPDATE.SEL NEWSEL CCH# NIL 'LEFT NIL TEMP))
                    elseif (ILESSP NCHLIM CCH#)
                      then ;; New is earlier
                           (CL:WHEN (SETQ TEMP (\TEDIT.FIND.PROTECTED.END TEXTOBJ CCH# (SUB1 NCHLIM)))
                               (\TEDIT.UPDATE.SEL NEWSEL CCH# NIL 'LEFT NIL TEMP)))
                  (SETQ NCH# (FGETSEL NEWSEL CH#))
                  (SETQ NCHLIM (FGETSEL NEWSEL CHLIM)))
              ;; There are now no protected characters between NEWSEL and CURSEL.
              ;;
              ;; NEWSEL's ONFLG will be T if we think that a currently highlighted region needs to be turned off, NIL if it needs to be turned on. I.e.
              ;; we show it with (NOT ONFLG).
              (FSETSEL NEWSEL ONFLG NIL)                          ; Should always be off at start but...
              (if (IGEQ NCH# CCHLIM)
                  then ;; CCC...NNN: NEWSEL after. Make CURSEL run from CCH# to NCHLIM, NEWSEL run from (ADD1 CCHLIM) to (SUB1
                       ;; NCH#)
                       (\TEDIT.UPDATE.SEL CURSEL CCH# NIL 'RIGHT NIL NCHLIM)
                       (\TEDIT.UPDATE.SEL NEWSEL CCHLIM NIL 'RIGHT NIL NCHLIM)
                elseif (ILEQ NCHLIM CCH#)
                  then ;; NNN...CCC: CURSEL before.  Make CURSEL run from NCH# to CCHLIM, NEWSEL run from NCH# to (SUB1 CCH#)
                       (\TEDIT.UPDATE.SEL CURSEL NCH# NIL 'LEFT NIL CCHLIM)
                       (\TEDIT.UPDATE.SEL NEWSEL NCH# NIL 'LEFT NIL CCH#)
                elseif (IGREATERP (IABS (IDIFFERENCE NCHLIM CCHLIM))
                                  (IABS (IDIFFERENCE NCH# CCH#)))
                  then ;; CCN[N|C]NC  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.
                       (\TEDIT.UPDATE.SEL CURSEL NCH# NIL 'LEFT NIL CCHLIM)
                       (FSETSEL NEWSEL ONFLG T)                   ; So highlighting turns off in the reduction
                       (\TEDIT.UPDATE.SEL NEWSEL CCH# NIL 'LEFT NIL NCH#)
                  else ;; O[O|N]NOOMust be shrinking from the right.
                       (\TEDIT.UPDATE.SEL CURSEL CCH# NIL 'RIGHT NIL NCHLIM)
                       (FSETSEL NEWSEL ONFLG T)
                       (\TEDIT.UPDATE.SEL NEWSEL NCHLIM NIL 'RIGHT NIL CCHLIM))
              (CL:UNLESS (EQ (FGETSEL CURSEL SELOBJ)
                             (FGETSEL NEWSEL SELOBJ))             ; Keep object if it is in overlapping part?
                  (FSETSEL CURSEL SELOBJ NIL))
              ;; NEWSEL now describes the difference between the highlighting of the original CURSEL to the highlighting of the extended CURSEL,
              ;; either putting up new highlighting or taking down old highlighting.
              (\TEDIT.FIXSEL NEWSEL TEXTOBJ)
              (\TEDIT.SHOWSEL NEWSEL (NOT (FGETSEL NEWSEL ONFLG))
                     TEXTOBJ)
              (FSETSEL NEWSEL ONFLG NIL)                          ; Restore its generally off state.
              (\TEDIT.FIXSEL CURSEL TEXTOBJ)))])
```

(\TEDIT.SCAN.LINE

```
  [LAMBDA (LINE X NEWSEL SELOPERATION TEXTOBJ BUTTON WORDSELFLG) ; Edited 18-Feb-2025 22:04 by rmk
                                                                 ; Edited 14-Feb-2025 09:47 by rmk
                                                                 ; Edited  3-Feb-2025 09:31 by rmk
                                                                 ; Edited  6-Dec-2024 11:06 by rmk
                                                                 ; Edited 30-Nov-2024 09:52 by rmk
                                                                 ; Edited 21-Oct-2024 00:07 by rmk
                                                                 ; Edited  6-Sep-2024 00:07 by rmk
                                                                 ; Edited  1-Aug-2024 17:13 by rmk
                                                                 ; Edited 20-Jun-2024 11:36 by rmk
                                                                 ; Edited 29-Apr-2024 12:35 by rmk
                                                                 ; Edited 15-Mar-2024 19:22 by rmk
                                                                 ; Edited 27-Jan-2024 23:44 by rmk
                                                                 ; Edited 14-Oct-2023 10:46 by rmk
                                                                 ; Edited  9-Apr-2023 18:21 by rmk
                                                                 ; Edited 31-May-91 12:26 by jds
```

```
;; Find the selection in LINE picked out by the mouse X coordinate.  This may run to the right if the mouse-position is protected. This also expands
;; to word selection in the current line, avoiding protected characters.
;;
;; Earlier versions had more complexity because it not ony figured out the character pointed at but also "fixed" the selection on the fly to avoid the
;; more generic \TEDIT.FIXLINE.The generic fixline would scan through the lines of a tall window to find the line containing the selected CH#, and
;; then apply \TEDIT.CHTOX to scan its (presumably cached) THISLINE to set up the X0 and XLIM. But not a noticeable delay for user
;; interaction--not worth the complexity.
(LINEDESCRIPTOR! LINE)
(TEXTOBJ! TEXTOBJ)
(SELECTION! NEWSEL)
(FSETSEL NEWSEL SET NIL)
(PROG (CHARSLOT CLOOKS CHNO X0 XLIM SELCHAR PASTRIGHT THISLINE MOVED)
      (SETQ THISLINE (FGETTOBJ TEXTOBJ THISLINE))
      (CL:UNLESS (EQ LINE (fetch DESC of THISLINE))            ; Make sure the cache describes this line
          (SETQ LINE (\TEDIT.FORMATLINE TEXTOBJ (FGETLD LINE LCHAR1)
                            LINE)))                            ; Convert X's display units to LINE's scale
      (SETQ XLIM (FGETLD LINE LX1))                            ; Pretend the "last" character ended at the margin
      (SETQ X (IMAX X XLIM))
      (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)))
                                                               (* (* ; "If not more than 30 past the end, put it inside the last
                                                               character.") (CL:WHEN (IGREATERP
                                                               (IDIFFERENCE X (FGETLD LINE LXLIM)) 30)
                                                               (RETURN NIL)))
          (SETQ X (SUB1 (FGETLD LINE LXLIM))))
      [SETQ CHARSLOT (for CS incharslots THISLINE do (if CHAR
                                                         then (add XLIM CHARW)
                                                         ; Start of the next character
                                                          (CL:WHEN (IGEQ XLIM X)
                                                                (RETURN CS))
                                                          (add CHNO 1)
                                                       else (SETQ CLOOKS CHARW]
                                                         ; The running CHARLOOKS
      (CL:UNLESS CHARSLOT                                      ; Guardrail
          (RETURN))
      (CL:WHEN (FGETCLOOKS CLOOKS CLPROTECTED)
          ;; Extensions can't run through protected characters, and they can't be deleted.
          (CL:WHEN (OR (EQ BUTTON 'RIGHT)
                       (EQ SELOPERATION 'DELETE))
                 (RETURN NIL))
          ;; Otherwise, if either CLSELAFTER or CLSELBEFORE, we move CHARSLOT, CHNO, XLIM,CLOOKS to the closest unprotected
          ;; one.
          [SETQ CHARSLOT (if (FGETCLOOKS CLOOKS CLSELAFTER)
                             then (SETQ MOVED 'FORWARD)
                                  (for CS incharslots (NEXTCHARSLOT CHARSLOT)
                                     do (if CHAR
                                            then (add XLIM CHARW)
                                                 (add CHNO 1)
                                                 (CL:UNLESS (FGETCLOOKS CLOOKS CLSELAFTER)
                                                       (RETURN CS))
                                          else (SETQ CLOOKS CHARW)))
                           elseif (FGETCLOOKS CLOOKS CLSELBEFORE)
                             then
                                  ;; We back up through the charlooks keeping track of the next previous CLSELBEFORE
                                  ;; (BEFORESLOT) while we look for the first one that is not CLSELBEFORE.  When we find that one,
                                  ;; we know that the PREVSLOT of BEFORESLOT is the one we want.
                                  (SETQ MOVED 'BACKWARD)
                                  (for CS (BEFORECHNO _ CHNO)
                                      (BEFOREX _ XLIM)
                                      (BEFORELOOKSLOT _ CHARSLOT)
                                      backcharslots
                                      (PREVCHARSLOT CHARSLOT) do (if CHAR
                                                                     then (add XLIM (IMINUS CHARW))
                                                                          (add CHNO -1)
                                                                   elseif (FGETCLOOKS CHARW CLSELBEFORE)
                                                                     then (SETQ BEFORECHNO CHNO)
                                                                          (SETQ BEFORELOOKSLOT CS)
                                                                          (SETQ BEFOREX XLIM)
                                                                   elseif BEFORELOOKSLOT
                                                                     then (SETQ XLIM BEFOREX)
                                                                          (SETQ CHNO BEFORECHNO)
                                                                          (RETURN (PREVCHARSLOT
                                                                                       BEFORELOOKSLOT])
      (CL:UNLESS CHARSLOT                                      ; Everything was protected.
          (RETURN))
;; CHNO and CHARSLOT:  the character pointed to, X0 is the beginning of CHNO, XLIM the point after CHNO.
      (SETQ SELCHAR (CHAR CHARSLOT))
;; NOTE:  This preserves the HOW and HOWHEIGHT fields as set by the caller
```

```
        (FSETSEL NEWSEL SELKIND 'CHAR)
        (FSETSEL NEWSEL X0 (IDIFFERENCE XLIM (CHARW CHARSLOT)))
                                                        ; Setting X0 suppresses an extra scan in FIXSEL
        (FSETSEL NEWSEL XLIM XLIM)
        (FSETSEL NEWSEL CH# CHNO)
        (FSETSEL NEWSEL SELOBJ NIL)
```

;; DCH=0 makes it a point selection, 1 picks out a single char. Original code produced 0 only for protected text and dummy lines. For copy/delete
;; selections, it's more convenient still to select at least one character, at least until modern one-button swiping is implemented.

;; If we end up in a protected piece, we want DCH=0. We then want to flash the caret iff we moved forward or backward

```
        [if (FGETCLOOKS CLOOKS CLPROTECTED)
            then (FSETSEL NEWSEL DCH 0)                 ; Protected: nothing shows
                 [FSETSEL NEWSEL HASCARET (AND MOVED (EQ SELOPERATION 'NORMAL]
          else (FSETSEL NEWSEL DCH (if (EQ 0 (TEXTLEN TEXTOBJ))
                                       then 0
                                     elseif (OR WORDSELFLG (NEQ SELOPERATION 'NORMAL)
                                                (FGETTOBJ TEXTOBJ TXTREADONLY)
                                                (type? IMAGEOBJ SELCHAR))
                                       then 1
                                       else 0           ; 0 = point selection, character not underlined.  But extension is
                                                        ; confusing
                                            1))
               (FSETSEL NEWSEL HASCARET (EQ SELOPERATION 'NORMAL]
        (FSETSEL NEWSEL CHLIM (IPLUS (FGETSEL NEWSEL CH#)
                                     (FGETSEL NEWSEL DCH)))
        (FSETSEL NEWSEL POINT (if (EQ (CHARCODE EOL)
                                      (CHAR CHARSLOT))
                                  then ;; Always go to the left of an EOL, so caret stays on its line
                                       'LEFT
                                elseif [OR PASTRIGHT (EQ MOVED 'BACKWARD)
                                           (AND (IGEQ (CHARW CHARSLOT)
                                                      3)
                                                (IGEQ X (IDIFFERENCE XLIM (FOLDLO (CHARW CHARSLOT)
                                                                                  2]
                                  then ;; Beyond the line, or towards the end of a character that is at least 3 points wide.
                                       'RIGHT
                                  else 'LEFT))          ; Don't recognize an object that wasn't directly pointed at
        (FSETSEL NEWSEL SELOBJ (CL:UNLESS PASTRIGHT (IMAGEOBJP SELCHAR)))
        (FSETSEL NEWSEL SET T)
```

;; Single-char selection is good

;;

```
        (CL:WHEN WORDSELFLG
```

;; Expand the (unprotected) selection to its word boundaries. This is done here because it makes use of THISLINE and CHARSLOT

```
            (CL:UNLESS (OR (FGETSEL NEWSEL SELOBJ)
                           (FGETLD LINE LDUMMY)
                           (FGETCLOOKS CLOOKS CLPROTECTED))
                       (\TEDIT.SCAN.LINE.WORD X TEXTOBJ THISLINE NEWSEL CHARSLOT CLOOKS)))
                                                        ; We now have a complete char/caret selection
        (RETURN NEWSEL])
```

(\TEDIT.SCAN.LINE.WORD

```
  [LAMBDA (X TEXTOBJ THISLINE NEWSEL CHARSLOT SELLOOKS)          ; Edited  7-Nov-2024 21:50 by rmk
                                                                 ; Edited  4-Oct-2024 08:39 by rmk
                                                                 ; Edited 28-Aug-2024 10:22 by rmk
                                                                 ; Edited  3-Aug-2024 12:41 by rmk
                                                                 ; Edited 16-Jul-2024 19:53 by rmk
                                                                 ; Edited 13-Jul-2024 10:39 by rmk
                                                                 ; 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
```

;; NEWSEL is a character selection at the CHARSLOT character in THISLINE. This expands it to its surrounding word boundaries. Looks are
;; tracked for protection. X is for accurate adjustment of the point--closest to the mouse X even if that's on the other edge of a wide character.

;;

;; This uses X at the original point granularity and CHARSLOT to do a more precise determination of the POINT in a word with wide characters.
;; Otherwise it would have to scan THISLINE to find the pivot character, so that it can savoid protected regions when scanning backward and
;; forward.

```
  (SELECTION! NEWSEL)
  (CL:UNLESS (EQ 'CHAR (FGETSEL NEWSEL SELKIND))
         (\TEDIT.THELP "Can only expand CHAR selections to WORD selections"))
  (LET ((CH# (FGETSEL NEWSEL CH#))
        (CHLIM (FGETSEL NEWSEL CHLIM))
        (X0 (FGETSEL NEWSEL X0))
        (XLIM (FGETSEL NEWSEL XLIM)))
```

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

```
        ;; CHLIM will become one past the last charno of the word selection
        ;; X0 will become the X at the beginning of the first char
        ;; XLIM will become the X at the end of last char                    ;
        (for CSLOT (CLOOKS _ SELLOOKS)
             (LASTCHAR _ (CHAR CHARSLOT))
             backcharslots
             (PREVCHARSLOT CHARSLOT) do (CL:UNLESS CHAR
                                            ;; CLOOKS is the looks AFTER the preceding char.  We have to go back further to see if the
                                            ;; current char is protected.
                                            (SETQ CLOOKS CHARW)
                                            (GO $$ITERATE))
                                        (CL:WHEN (OR (type? IMAGEOBJ CHAR)
                                                     (\TEDIT.WORD.BOUND TEXTOBJ CHAR LASTCHAR)
                                                     (fetch (CHARLOOKS CLPROTECTED) of CLOOKS))
                                                                    ; Stop at a protection bounary
                                            (RETURN))
                                        (SETQ LASTCHAR CHAR)
                                        (ADD X0 (IMINUS CHARW))
                                        (ADD CH# -1))
        ;; And search forward for the end of the word
        (for CSLOT (CLOOKS _ SELLOOKS)
             (PREVCHAR _ (CHAR CHARSLOT))
             incharslots
             (NEXTCHARSLOT CHARSLOT) do (CL:UNLESS CHAR
                                            (SETQ CLOOKS CHARW)
                                            (GO $$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 NEWSEL SELKIND 'WORD)
        (FSETSEL NEWSEL CH# CH#)
        (FSETSEL NEWSEL CHLIM CHLIM)
        (FSETSEL NEWSEL DCH (IDIFFERENCE CHLIM CH#))
        (FSETSEL NEWSEL X0 X0)
        (FSETSEL NEWSEL 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 NEWSEL POINT (if [OR (fetch (CHARLOOKS CLSELAFTER) of SELLOOKS)
                                      (AND (IGEQ (IDIFFERENCE XLIM X0)
                                                 3)
                                           (IGEQ X (FOLDLO (IPLUS XLIM X0)
                                                           2]
                                  then 'RIGHT
                                  else 'LEFT])
```

(**TEDIT.XYTOSEL**

```
  [LAMBDA (X Y NEWSEL TEXTOBJ SELOPERATION PANE BUTTON CURSEL REGIONTYPE)
                                                                        ; Edited 13-Feb-2025 11:03 by rmk
                                                                        ; Edited 17-Dec-2024 10:10 by rmk
                                                                        ; Edited  6-Dec-2024 12:00 by rmk
                                                                        ; Edited 30-Nov-2024 14:15 by rmk
                                                                        ; Edited 28-Nov-2024 14:39 by rmk
                                                                        ; Edited  7-Nov-2024 21:49 by rmk
                                                                        ; Edited  4-Oct-2024 07:57 by rmk
                                                                        ; Edited 31-Jul-2024 00:13 by rmk
                                                                        ; Edited 29-Apr-2024 12:33 by rmk
                                                                        ; Edited 15-Mar-2024 13:36 by rmk
                                                                        ; Edited 26-Dec-2023 08:53 by rmk
                                                                        ; Edited 20-Jul-2023 20:38 by rmk
                                                                        ; Edited 27-May-2023 15:18 by rmk
                                                                        ; Edited 31-May-91 12:26 by jds
    ;; Sets NEWSEL to the selection picked out by the X-Y coordinates in PANE. If unsuccessful, NEWSEL is unset.
    ;; CURSEL is used to decide whether extensions go to words or paragraphs (and to turn off highlighting for objects).
    (SELECTION! NEWSEL)
    (TEXTOBJ! TEXTOBJ)
    (FSETSEL NEWSEL SET NIL)
    (PROG (LINE PARAFIRSTCHNO PARALASTCHNO SELFN)
          (CL:UNLESS (SETQ LINE (\TEDIT.XYTOSEL.LINE X Y PANE TEXTOBJ))
                (RETURN))
          (SELECTQ (\TEDIT.REGIONTYPE BUTTON CURSEL TEXTOBJ REGIONTYPE)
               ((TEXT PANE)                                             ; We're in the regular text area, which character?
                    (CL:WHEN (AND (IGREATERP (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))))
                       (CL:WHEN (AND (\TEDIT.SCAN.LINE LINE X NEWSEL SELOPERATION TEXTOBJ BUTTON
                                     (SELECTQ BUTTON
                                          (RIGHT (MEMB (FGETSEL CURSEL SELKIND)
                                                       '(WORD PARA)))
                                          (MIDDLE T)
                                          NIL))
                                (FGETSEL NEWSEL SELOBJ)
                                CURSEL)
                            ;; Run the buttonin function with CURSEL's highlighting turned off--its highlighting may be somewhere else
                            (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                            (\TEDIT.SELECT.OBJECT TEXTOBJ NEWSEL LINE X Y PANE SELOPERATION BUTTON)))
                  (LINE (CL:WHEN (FGETTOBJ TEXTOBJ MENUFLG)
                             ;; Except for fields, menus are completely protected.  Confusing to deal with a field that spreads across several lines,
                             ;; so essentially disable the line region
                             (FSETSEL NEWSEL SET NIL)
                             (RETURN))
                        ;; FIXSEL deals with other panes.  LEFT because line-select area is on the left, but perhaps could be RIGHT if extending.
                        (\TEDIT.UPDATE.SEL NEWSEL (FGETLD LINE LCHAR1)
                               NIL
                               'LEFT NIL (FGETLD LINE LCHARLIM))     ; Not selecting an object just yet
                        (FSETSEL NEWSEL SELKIND 'LINE)
                        (FSETSEL NEWSEL SELOBJ NIL)
                        (FSETSEL NEWSEL SET T))
                  (PARA (CL:WHEN (FGETTOBJ TEXTOBJ MENUFLG)
                             (FSETSEL NEWSEL SET NIL)
                             (RETURN))
                        ;; Find its first and last character numbers, whether or not they are visible in any pane.  FIXSEL figures out  what's visible
                        ;; where
                        (FSETSEL NEWSEL SELOBJ NIL)
                        (SETQ PARAFIRSTCHNO (CAR (\TEDIT.PARA.FIRST TEXTOBJ (FGETLD LINE LCHAR1)
                                                                    T)))
                        (SETQ PARALASTCHNO (CAR (\TEDIT.PARA.LAST TEXTOBJ (FGETLD LINE LCHARLAST)
                                                                  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 NEWSEL PARAFIRSTCHNO NIL (CL:IF (ILEQ (IDIFFERENCE (FGETLD LINE LCHAR1)
                                                                                              PARAFIRSTCHNO)
                                                                                 (IDIFFERENCE (ADD1 PARALASTCHNO)
                                                                                              (FGETLD LINE LCHARLAST)))
                                                                           'LEFT
                                                                           'RIGHT)
                               NIL
                               (ADD1 PARALASTCHNO))
                        (FSETSEL NEWSEL SELOBJ NIL)
                        (FSETSEL NEWSEL SELKIND 'PARA)
                        (FSETSEL NEWSEL SET T))
                  (\TEDIT.THELP "Unknown text/line-bar region?"))
             (CL:UNLESS (FGETSEL NEWSEL SET)                        ; Invalid
                 (RETURN))
             (CL:WHEN [AND (SETQ SELFN (GETTEXTPROP TEXTOBJ 'SELFN))
                           (EQ 'DON'T (APPLY* SELFN TEXTOBJ NEWSEL SELOPERATION 'TENTATIVE]
                 ;; Declined by user function.
                 (FSETSEL NEWSEL SET NIL)
                 (RETURN))
             (FSETSEL NEWSEL ONFLG NIL)                             ; New selection not yet displayed
             (\TEDIT.FIXSEL NEWSEL TEXTOBJ)
             (RETURN NEWSEL])
```

(\TEDIT.REGIONTYPE

```
[LAMBDA (BUTTON CURSEL TEXTOBJ REGIONTYPE)                          ; Edited  6-Dec-2024 12:50 by rmk
    ;; Coerces the mouse region according to the button and current selection.
    (CL:UNLESS REGIONTYPE
        (SETQ REGIONTYPE (FGETTOBJ TEXTOBJ MOUSEREGION)))
    (SELECTQ BUTTON
         (MIDDLE (CL:WHEN (AND (EQ REGIONTYPE 'LINE)
                               (FGETTOBJ TEXTOBJ PARABREAKCHARS))
                        ;; A middle-button selection in the line region means the line-containing paragraph.  If there are no PARABREAKCHARS, we
                        ;; assume heuristically that there are no paragraphs, and a line is just a line.  Otherwise, a middle click in such a document
                        ;; will select the whole thing, not very useful.
                        (SETQ REGIONTYPE 'PARA)))
         (RIGHT (SETQ REGIONTYPE (OR [CAR (MEMB (FGETSEL CURSEL SELKIND)
                                                '(LINE PARA]
                                     'TEXT)))
         NIL)
```

```
    REGIONTYPE])
```

(\TEDIT.XYTOSEL.INLINEP

```
  [LAMBDA (X Y PANE TEXTOBJ)                                        ; Edited 30-Nov-2024 15:46 by rmk
    ;; If the last click points to a valid line in PANE, return that LINE.  Allow clicks slightly beyond the end.
    (LET ((LINE (\TEDIT.XYTOSEL.LINE X Y PANE TEXTOBJ)))
         (CL:WHEN (AND LINE (ILEQ (IDIFFERENCE X (FGETLD LINE LXLIM))
                                  30))
                LINE])
```

(\TEDIT.XYTOSEL.LINE

```
  [LAMBDA (X Y PANE TEXTOBJ)                                        ; Edited  1-Dec-2024 11:45 by rmk
                                                                    ; Edited 30-Nov-2024 10:08 by rmk
    ;; Which line did the click go down in?
    (AND (INSIDEP (PANEREGION PANE)
                  X Y)
         (for L inlines (FGETLD (PANEPREFIX PANE)
                                NEXTLINE)
            when (ILEQ (FGETLD L YBOT)
                       Y)
            do                                                      ; Don't select an empty line
               (RETURN (CL:IF (ZEROP (FGETLD L LNCH))
                              $$PREVLINE
                              L))
            finally ;; Y is below the last line.  Assume $$PREVLINE points to the last.  But maybe last is the end-of-document dummy
                    (CL:UNLESS (EQ L (PANESUFFIX PANE))
                        (RETURN (if (AND L (IGEQ (FGETLD $$PREVLINE LCHAR1)
                                                 (TEXTLEN TEXTOBJ)))
                                    then (FGETLD $$PREVLINE PREVLINE)
                                  else $$PREVLINE)))])

)

(DEFINEQ
```

(\TEDIT.FIXSEL

```
  [LAMBDA (SEL TEXTOBJ AVOIDPANE ONLYPANE)                          ; Edited  1-Dec-2024 11:28 by rmk
                                                                    ; Edited 28-Nov-2024 14:30 by rmk
                                                                    ; Edited 25-Nov-2024 12:57 by rmk
                                                                    ; Edited 19-Nov-2024 15:52 by rmk
                                                                    ; Edited 17-Nov-2024 15:58 by rmk
                                                                    ; Edited  3-Oct-2024 18:47 by rmk
                                                                    ; Edited  9-Sep-2024 09:26 by rmk
                                                                    ; Edited  3-Sep-2024 13:16 by rmk
                                                                    ; Edited  6-Jul-2024 22:36 by rmk
                                                                    ; Edited  4-Jul-2024 15:45 by rmk
                                                                    ; Edited 28-Jun-2024 21:50 by rmk
                                                                    ; Edited 24-Jun-2024 23:57 by rmk
                                                                    ; Edited 16-Jun-2024 22:02 by rmk
                                                                    ; Edited 21-May-2024 09:01 by rmk
                                                                    ; Edited 29-Apr-2024 12:56 by rmk
                                                                    ; Edited 26-Apr-2024 00:23 by rmk
                                                                    ; 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
    ;; The PANEPREFIX 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 being 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 PANEPREFIX 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.
    (CL:UNLESS (type? TEXTOBJ TEXTOBJ)
        (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ)))
    (CL:UNLESS SEL
        (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
    (SELECTION! SEL)
    (CL:WHEN (AND (FGETTOBJ TEXTOBJ PRIMARYPANE)
                  (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 (FIRSTCHNO _ (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 (PANETOPLINE PANE) first (RPLACA L1 NIL)
                                                                    (RPLACA LN NIL)
                                  suchthat
                                      ;; The first visible line in PANE that SEL selects.
                                      (FLINESELECTEDP L FIRSTCHNO LASTCHNO)))
           do ;; 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.  For interior lines, SHOWSEL.HILIGHT uses their LX1 and LXLIM values.
              ;; 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 (FWITHINLINEP FIRSTCHNO PSTARTLINE)
                      [SETQ X0 (\TEDIT.CHTOLINEX TEXTOBJ PSTARTLINE (IMAX FIRSTCHNO (FGETLD PSTARTLINE LCHAR1))
                                      (AND (IGREATERP FIRSTCHNO (TEXTLEN TEXTOBJ))
                                           (GETLD (FGETLD PSTARTLINE PREVLINE)
                                                  FORCED-END]
                      (FSETSEL SEL X0 X0)))
              [SETQ PENDLINE (for L (PBOTTOM _ (PANEBOTTOM PANE))
                                inlines PSTARTLINE do ;; Stop when L is beyond  the selection or below the screen.
                                                      (CL:WHEN (ILESSP 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))
              (CL:UNLESS XLIM
                  (CL:WHEN (FWITHINLINEP LASTCHNO PENDLINE)
                      (SETQ XLIM (\TEDIT.CHTOLINEX TEXTOBJ PENDLINE LASTCHNO T))
                      (FSETSEL SEL XLIM XLIM)))
              ;; Fill in the selected lines that are visible in this pane
              (RPLACA L1 PSTARTLINE)
              (RPLACA LN PENDLINE)))
    SEL])
```

(\TEDIT.CHTOLINEX

```
  [LAMBDA (TEXTOBJ LINE CH# AFTER)                     ; Edited  6-Mar-2025 11:57 by rmk
                                                       ; Edited 28-Nov-2024 14:41 by rmk
                                                       ; Edited 17-Nov-2024 15:58 by rmk
                                                       ; Edited 13-Jun-2024 17:12 by rmk
                                                       ; Edited 10-May-2024 00:26 by rmk
                                                       ; 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:UNLESS (EQ LINE (fetch DESC of THISLINE))
             ;; Reformat if LINE is not cached in THISLINE.
             (\TEDIT.FORMATLINE (FGETTOBJ TEXTOBJ STREAMHINT)
                    (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 LCHARLAST)))
             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 eachtime (CL:WHEN (AND CHAR (DIACRITICP CHAR))
                                                        ;; If the diacritic CHARW is greater than the CHARW of the next slot, we should set the diacritic
                                                        ;; CHARW to (IDIFFERENCE CHARW (NEXT CHARW)).
                                                           ;; i.e. (IMAX 0 (IDIFFERENCE CHARW (NEXT CHARW))
                                                        (SETQ CHARW 0))
                         unless (type? CHARLOOKS CHARW) do ;; Update the running X-position in the line, skiping look-slots and skipping diacritics
                                                        (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))
                                               (IGREATERP (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)))])

)

(DEFINEQ

(\TEDIT.RESET.EXTEND.PENDING.DELETE
  [LAMBDA (TEXTOBJ)                                          ; Edited 26-Nov-2024 23:44 by rmk
                                                             ; 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
    (\TEDIT.SET.SEL.LOOKS (TEXTSEL TEXTOBJ)
           'NORMAL)
    (SETTOBJ TEXTOBJ BLUEPENDINGDELETE NIL])

(\TEDIT.SET.SEL.LOOKS
  [LAMBDA (SEL OPERATION)                                    ; Edited 28-Feb-2025 17:45 by rmk
                                                             ; Edited  7-Nov-2024 21:50 by rmk
                                                             ; Edited  4-Oct-2024 08:40 by rmk
                                                             ; 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))
         (NIL)
         (\TEDIT.THELP "UNKNOWN SELECTION OPERATION" OPERATION))
    SEL])

)

(DEFINEQ
```

(**TEDIT.SHOWSEL**

```
  [LAMBDA (SEL ON TEXTOBJ ONLYPANE)                                 ; Edited  4-Oct-2024 10:29 by rmk
                                                                    ; Edited  2-Oct-2024 14:20 by rmk
                                                                    ; Edited 21-Aug-2024 16:11 by rmk
                                                                    ; Edited 19-Jul-2024 23:46 by rmk
                                                                    ; Edited 18-Jul-2024 12:14 by rmk
                                                                    ; Edited  6-Jul-2024 22:44 by rmk
                                                                    ; Edited 28-Jun-2024 22:34 by rmk
                                                                    ; Edited 18-May-2024 19:56 by rmk
                                                                    ; Edited 29-Apr-2024 13:01 by rmk
                                                                    ; Edited  9-Mar-2024 12:01 by rmk
                                                                    ; Edited 20-Mar-2024 10:56 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
    (CL:WHEN (TEXTSTREAMP TEXTOBJ)
        (SETQ TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TEXTOBJ)))
    (TEXTOBJ! TEXTOBJ)
    (CL:UNLESS SEL
        (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
    (SELECTION! SEL)
```

;; 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 (AND (FGETSEL SEL SET)
                  (NEQ ON (FGETSEL SEL ONFLG))
                  (NOT (FGETTOBJ TEXTOBJ TXTDON'TUPDATE))
                  (FGETTOBJ TEXTOBJ PRIMARYPANE))
```

;; This operation only makes sense if the selection is set, it is not currently in the intended ON state, we are allowed to update, and there is at ;; least one pane to highlight in.

```
        (if (FGETSEL SEL SELOBJ)
            then
```

;; SELOBJ if the selection consisted only of a single image object. It presumably did its own buttonevent operations when it was ;; selected, but is otherwise immune to normal highlighting. But it acts just as a normal character in all panes if it is part of a longer ;; selection.

;; This does the WHENOPERATEDONFN once no matter how many panes the object appears in, and that function controls the ;; highlighting. Not sure what happens in other panes. If we do the ordinary highlighting, then e.g. a whole NWAY image object ;; gets underlines, even though only one toggle was selected.

```
                (\TEDIT.OPERATE.OBJECT (FGETTOBJ TEXTOBJ STREAMHINT)
                       SEL
                       (FGETTOBJ TEXTOBJ SELPANE)
                       (CL:IF ON
                           'HIGHLIGHTED
                           'UNHIGHLIGHTED))
          else (for PANE inpanes (PROGN TEXTOBJ) as L1 in (FGETSEL SEL L1) as LN in (FGETSEL SEL LN)
                  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)))
        (FSETSEL SEL ONFLG ON))])
```

(**TEDIT.SHOWSEL.HILIGHT**

```
  [LAMBDA (TEXTOBJ L1 LN PANE SEL)                                  ; Edited  1-Dec-2024 11:28 by rmk
```

```
                                                                    ; Edited 20-Nov-2024 10:21 by rmk
                                                                    ; Edited 28-Oct-2024 13:57 by rmk
                                                                    ; Edited  4-Oct-2024 08:09 by rmk
                                                                    ; Edited 12-Sep-2024 20:47 by rmk
                                                                    ; Edited  6-Sep-2024 11:07 by rmk
                                                                    ; Edited 13-Jun-2024 22:04 by rmk
                                                                    ; 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.

    (LINEDESCRIPTOR! L1)
    (LINEDESCRIPTOR! LN)

    ;; If the first visible line (L1) is also the first line of the selection (it contains CH#), then the highlight's left-boundary is SEL's X0, else the line's LX1.
    ;; Similarly for the last visible LN/XLIM/LXLIM.

    (for L LEFT RIGHT DISTBELOW (CH# _ (FGETSEL SEL CH#))
        (LASTCHNO _ (SUB1 (FGETSEL SEL CHLIM)))
        (SHADE _ (OR (FGETSEL SEL HOW)
                     BLACKSHADE))
        (SHADEHEIGHT _ (OR (FGETSEL SEL HOWHEIGHT)
                           1))
        (PBOTTOM _ (PANEBOTTOM PANE)) first
                                        ;; DISTBELOW=1 would give a 1-pt spacing between the line-bottom and the selection underline, so
                                        ;; that the line doesn't go through the last point of a descender.  If 0, the slection line runs through the
                                        ;; bottom of an image object, makes 1-point horizontal rules invisible and the selection line may not be
                                        ;; so obvious for other image objects.  However:  1 has to be coordinated with the scroll-up functions:
                                        ;; If the highlight is below the descender of the last line in a pane, the highlighting will not move up
                                        ;; when the line is bltted--it disappears. that the highlight on the bottom line moves up when the line
                                        ;; itself is bltted.  The highlighting would be 1 point below the bottom of the line.

                                        ;; At least some of the affected code can be located by (. WHO USES DISTBELOW). Setting it to 1
                                        ;; here and there still doesn't give the righ effects.  An alternative might be to say that DESCENT has
                                        ;; a 1-point minimum.  Not sure what that would do to interline spacing.  Maybe also subtract it from
                                        ;; the line leading?  TBD
            (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 (CL:IF (WITHINLINEP CH# L)
                         (FGETSEL SEL X0)
                         (FGETLD L LX1)))
       (SETQ RIGHT (CL:IF (WITHINLINEP LASTCHNO L)
                          (FGETSEL SEL 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.SEL**

```
  [LAMBDA (SEL CH# DCH POINT LOOKS CHLIM)                           ; Edited 10-Jul-2024 17:25 by rmk
                                                                    ; Edited  8-Jul-2024 00:11 by rmk
                                                                    ; Edited 21-Jun-2024 14:21 by rmk
                                                                    ; Edited 29-Apr-2024 13:28 by rmk
                                                                    ; 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.

    ;; For convenience, If DCH is NIL and CHLIM is provided, DCH is computed from CH# and CHLIM instead of being left alone.

    [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 (if CHLIM
                           then (IDIFFERENCE CHLIM CH#)
                         else (FGETSEL SEL DCH))))
         (CL:UNLESS POINT
             (SETQ POINT (FGETSEL SEL POINT)))]
    ;; If below 1, left of 1.  We don't know TEXTLEN without the TEXTOBJ, so we can't test the length.
    (CL:WHEN (ILESSP CH# 1)
        (SETQ CH# 1)
        (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).
    (SETSEL SEL CH# CH#)
    (FSETSEL SEL DCH DCH)
    (FSETSEL SEL CHLIM (CL:IF (EQ DCH 0)
                              (ADD1 CH#)
                              (IPLUS CH# DCH)))
    (FSETSEL SEL POINT POINT)
    (FSETSEL SEL SELOBJ NIL)                                    ; If we are moving around, we are moving away from any old
                                                                ; object
    (FSETSEL SEL SET T)
    (CL:WHEN LOOKS (\TEDIT.SET.SEL.LOOKS SEL LOOKS))
    SEL])
```

(\TEDIT.CARETLINE

```
  [LAMBDA (SEL PANE TEXTOBJ)                                    ; Edited  7-Nov-2024 21:50 by rmk
                                                                ; Edited  4-Oct-2024 08:40 by rmk
                                                                ; Edited 24-Apr-2024 11:33 by rmk
    ;; Returns the line in PANE that contains SEL's caret (NIL if the caret isn't showing in PANE).
    (CL:WHEN (GETSEL SEL SET)
        (SELECTQ (GETSEL SEL POINT)
            (LEFT (\TEDIT.SEL.L1 SEL PANE TEXTOBJ))
            (RIGHT (\TEDIT.SEL.LN SEL PANE TEXTOBJ))
            (\TEDIT.THELP "ILLEGAL POINT" (GETSEL SEL POINT)))))])
```

(\TEDIT.SEL.L1

```
  [LAMBDA (SEL PANE TEXTOBJ)                                    ; Edited 24-Apr-2024 08:34 by rmk
                                                                ; Edited  8-Apr-2024 23:42 by rmk
                                                                ; Edited 16-Nov-2023 23:43 by rmk
    ;; Returns L1 for PANE in SEL
    (for L in (GETSEL SEL L1) as P inpanes (PROGN TEXTOBJ) when (EQ P PANE) do (RETURN L])
```

(\TEDIT.SEL.LN

```
  [LAMBDA (SEL PANE TEXTOBJ)                                    ; Edited 24-Apr-2024 08:34 by rmk
                                                                ; Edited  8-Apr-2024 23:41 by rmk
                                                                ; Edited 16-Nov-2023 23:43 by rmk
    ;; Returns LN for PANE in SEL
    (for L in (GETSEL SEL LN) as P inpanes (PROGN TEXTOBJ) when (EQ P PANE) do (RETURN L])
```

(\TEDIT.SEL.DELETEDCHARS

```
  [LAMBDA (SELTOFIX FIRSTCHAR LEN)                              ; Edited  6-Feb-2025 15:53 by rmk
                                                                ; Edited  4-Feb-2025 23:05 by rmk
                                                                ; Edited 26-Nov-2024 22:31 by rmk
                                                                ; Edited  7-Jul-2024 12:09 by rmk
                                                                ; 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 LEN characters starting at FIRSTCHAR have been or will be removed.
    (CL:WHEN (type? SELECTION FIRSTCHAR)
        (SETQ LEN (FGETSEL FIRSTCHAR DCH))
        (SETQ FIRSTCHAR (FGETSEL FIRSTCHAR CH#)))
    (LET ((LASTCHAR (IPLUS FIRSTCHAR LEN -1))
          (B (FGETSEL SELTOFIX CH#))
          (E (FGETSEL SELTOFIX CHLAST))
          (DCH (FGETSEL SELTOFIX DCH)))
         ;; No overlap
         ;;  1          FddL   F gt E
         ;;       B23E              nothing
         ;;  2   FddL          L lt B
         ;;             B123E        B=B - LEN
```

```
          ;; Overlaps:  NEWB=(MIN F B)  = X+1  NEWDCH = (IMAX 0, E-L)
          ;;   3  XFddL              F leq B     L lt E
          ;;        X  [B23]45E              45E at F  DCH=E-L   X45E
          ;;        X45E        E-L
          ;;   4  XFddddddddL       F leq B       L geq E
          ;;        X[   B234E]
          ;;        X                    point selection at F DCH=0    E-L lt 0   DCH-LEN < 0
          ;;   5    X     FddL          F geq B   L lt E
          ;;        XB2[3456]7E
          ;;        XB27E               B27E at B  DCH = DCH - LEN
          ;;   6     X      FddL        F geq B    L geq  E
          ;;         XB2[3E
          ;;         XB2                B2 at B
          (if (IGREATERP FIRSTCHAR E)
              then                                                    ; Case 1: Nothing
                   NIL
            elseif (ILESSP LASTCHAR B)
              then                                                    ; Case 2: move back
                   (add (FGETSEL SELTOFIX CH#)
                        (IMINUS LEN))
                   (add (FGETSEL SELTOFIX CHLIM)
                        (IMINUS LEN))
              else                                                    ; Overlaps
                   (\TEDIT.UPDATE.SEL SELTOFIX (IMIN B FIRSTCHAR)
                          [if (ILEQ FIRSTCHAR B)
                              then                                    ; Cases 3 4
                                   (IMAX 0 (IDIFFERENCE E LASTCHAR))
                            elseif (ILEQ LASTCHAR E)
                              then                                    ; Case 5
                                   (IDIFFERENCE DCH LEN)
                              else                                    ; Case 6
                                  (IDIFFERENCE DCH (ADD1 (IDIFFERENCE E FIRSTCHAR]
                                 'LEFT])

)

(DEFINEQ

(\TEDIT.COPYSEL
  [LAMBDA (FROM TO)                                                   ; Edited  3-Sep-2024 22:44 by rmk
                                                                      ; Edited  7-Jul-2024 11:21 by rmk
                                                                      ; Edited 30-Jun-2024 23:21 by rmk
                                                                      ; Edited 29-Apr-2024 12:35 by rmk
                                                                      ; 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
    ;; This copies FROM to TO, but does not include the SELTEXTSTREAM.
    (\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 SELLINES (COPY (FGETSEL FROM SELLINES)))
             (FSETSEL TO POINT (FGETSEL FROM POINT))
             (FSETSEL TO SET (FGETSEL FROM SET))
             (FSETSEL TO SELTEXTSTREAM NIL)
             (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 SELTEXTSTREAM _ NIL L1 _ (COPY (FGETSEL FROM L1))
                                            LN _ (COPY (FGETSEL FROM LN))
                                            SELLINES _ (COPY (FGETSEL FROM SELLINES]
    TO])

(\TEDIT.SEL.CHANGED?
  [LAMBDA (NEWSEL OLDSEL)                                             ; Edited 19-Oct-2024 12:15 by rmk
                                                                      ; Edited 11-Sep-2024 00:02 by rmk
```

; Edited 18-Jul-2024 08:24 by rmk
; Edited 10-Jul-2024 16:27 by rmk
; Edited 8-Jul-2024 23:23 by rmk
; Edited 7-Jul-2024 08:58 by rmk
; Edited 29-Apr-2024 13:00 by rmk
; 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.

```
(SELECTION! NEWSEL)
(SELECTION! OLDSEL)
(NOT (AND (FGETSEL OLDSEL SET)
          (IEQP (FGETSEL NEWSEL CH#)
                (FGETSEL OLDSEL CH#))
          (IEQP (FGETSEL NEWSEL CHLIM)
                (FGETSEL OLDSEL CHLIM))
          (IEQP (FGETSEL NEWSEL DCH)
                (FGETSEL OLDSEL DCH))
          (EQ (FGETSEL NEWSEL POINT)
              (FGETSEL OLDSEL POINT))
          (EQ (FGETSEL NEWSEL HOW)
              (FGETSEL OLDSEL HOW))
          (EQ (FGETSEL NEWSEL HOWHEIGHT)
              (FGETSEL OLDSEL HOWHEIGHT))
          (EQ (FGETSEL NEWSEL SELKIND)
              (FGETSEL OLDSEL SELKIND))
          (EQ (FGETSEL NEWSEL HASCARET)
              (FGETSEL OLDSEL HASCARET))
          (EQ (FGETSEL NEWSEL SELOBJ)
              (FGETSEL OLDSEL SELOBJ])

)
```

;; Image objects

```
(DEFINEQ
```

(\TEDIT.SELECT.OBJECT

```
  [LAMBDA (TEXTOBJ NEWSEL LINE X Y SELPANE SELOPERATION BUTTON)
```

; Edited 6-Dec-2024 11:09 by rmk
; Edited 30-Nov-2024 00:01 by rmk
; Edited 26-Nov-2024 03:45 by rmk
; Edited 19-Oct-2024 20:09 by rmk
; Edited 5-Oct-2024 22:45 by rmk
; Edited 21-Aug-2024 15:29 by rmk
; Edited 20-Aug-2024 10:12 by rmk
; Edited 26-Jul-2024 14:31 by rmk
; Edited 20-Jul-2024 09:16 by rmk
; Edited 18-Jul-2024 12:17 by rmk
; Edited 13-Jun-2024 17:11 by rmk
; Edited 24-Apr-2024 09:50 by rmk
; 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

```
    (LET ((OBJ (FGETSEL NEWSEL SELOBJ))
          RESULT)
         (RESETLST
             (\TEDIT.CLIP.OBJECT OBJ (FGETSEL NEWSEL X0)
                    LINE PANE)                                  ; Go tell him he's being pointed at.
                                                                ; Note: SELOPERATION is undocumented
             (SETQ RESULT (ERSETQ (APPLY* (IMAGEOBJPROP OBJ 'BUTTONEVENTINFN)
                                          OBJ
                                          (WINDOWPROP SELPANE 'DSP)
                                          NEWSEL
                                          (IDIFFERENCE X (FGETSEL NEWSEL X0))
                                          (IDIFFERENCE Y (FGETLD LINE YBASE))
                                          SELPANE
                                          (fetch (TEXTWINDOW WTEXTSTREAM) of SELPANE)
                                          BUTTON SELOPERATION))))
```

;; The clipping region is now restored.

```
         (if (OR (NULL RESULT)
                 (EQMEMB 'DON'T RESULT)
                 (EQMEMB 'DONT RESULT))
             then
```

;; If NIL, an error happened. Maybe we should propagate the error, e.g. RETFROM the BUTTONEVENTFN?

;; If DON't the object declines to be selected.

```
              (FSETSEL NEWSEL SET NIL)
           elseif (EQMEMB 'CHANGED RESULT)
             then
```

;; The object may have updated its own image, within its coordinate system. But its box may have changed, and if so, the
;; document also needs to reformat and the selection has to be adjusted. We know that CURSEL is currently displayed, we get it
;; out of the way here, expecting that \TEDIT.BUTTONEVENTFN will synchronize CURSEL with NEWSEL.

```
                (\TEDIT.UPDATE.LINES TEXTOBJ 'CHANGED (FGETSEL NEWSEL CH#)
                   1)
                (\TEDIT.SHOWSEL NIL T TEXTOBJ)
                (FSETTOBJ TEXTOBJ \DIRTY T)
          elseif (NULL (CAR RESULT))
            then ;; NIL: Highlighting and selection have been taken care of, nothing for Tedit to do on this object.
                (FSETSEL NEWSEL SELOBJ NIL)
          else ;; Only non-NIL appears to matter:  could be just T ?
               (CAR RESULT])
```

(\TEDIT.SHOWSEL.OBJECT

```
  [LAMBDA (TEXTOBJ SEL L1 ON PANE)                                  ; Edited  1-Dec-2024 11:52 by rmk
                                                                    ; Edited 21-Aug-2024 15:31 by rmk
                                                                    ; Edited 19-Jul-2024 23:15 by rmk
                                                                    ; Edited 18-Jul-2024 12:19 by rmk
                                                                    ; Edited 24-Jan-2024 09:27 by rmk
                                                                    ; Edited 25-Nov-2023 15:48 by rmk
                                                                    ; Edited 14-Oct-2023 12:12 by rmk
                                                                    ; Edited  6-Jun-2023 15:28 by rmk
                                                                    ; Edited  1-May-2023 14:36 by rmk
                                                                    ; Edited  9-Apr-2023 15:37 by rmk
                                                                    ; Edited 12-Jun-90 17:50 by mitani
    ;; We are hilighting (or dehilighting) a selection containing only a single image object if it appears in PANE
    (LET [(OBJ (FGETSEL SEL SELOBJ))
          (IMAGEFN (IMAGEOBJPROP (FGETSEL SEL SELOBJ)
                                 'WHENOPERATEDONFN]
         (CL:WHEN (AND IMAGEFN (INSIDE? (PANEREGION PANE)
                                        (FGETSEL SEL X0)
                                        (FGETLD L1 YBOT)))
             ;; The selection is in the pane and has an image function
             (RESETLST
                 (\TEDIT.CLIP.OBJECT OBJ (FGETSEL SEL X0)
                        L1 PANE)
                 (ERSETQ (APPLY* IMAGEFN OBJ PANE (CL:IF ON
                                                         'HIGHLIGHTED
                                                         'UNHIGHLIGHTED)
                                 SEL
                                 (FGETTOBJ TEXTOBJ STREAMHINT)))))])
```

(\TEDIT.CLIP.OBJECT

```
  [LAMBDA (OBJ X LINE PANE)                                         ; Edited  1-Dec-2024 11:54 by rmk
                                                                    ; Edited 21-Aug-2024 15:38 by rmk
                                                                    ; Edited 19-Jul-2024 18:21 by rmk
```

;; Resets the coordinate system of PANE to the coordinate system of the object. Original PANE coordinates are restored when an enclosing ;; RESETLST is exited.

;; The 0,0 is the location of the bottom-left corner of the object, (width,height) is the upper-right corner of the object.

;; Gets the object's PANE Y position from LINE, X position from SEL.

```
    (LET [(OBJBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
                      (IMAGEBOX OBJ PANE)))
          (DS (WINDOWPROP PANE 'DSP]
         [RESETSAVE (DSPXOFFSET (IDIFFERENCE (IPLUS X (DSPXOFFSET NIL DS))
                                             (fetch XKERN of OBJBOX))
                                DS)
                '(PROGN (DSPXOFFSET OLDVALUE ,DS]
         [RESETSAVE (DSPYOFFSET (IDIFFERENCE (IPLUS (GETLD LINE YBASE)
                                                    (DSPYOFFSET NIL DS))
                                             (fetch YDESC of OBJBOX))
                                DS)
                '(PROGN (DSPYOFFSET OLDVALUE ,DS]
         (RESETSAVE (DSPCLIPPINGREGION (create REGION
                                               LEFT _ 0
                                               BOTTOM _ 0
                                               WIDTH _ (IMIN (fetch XSIZE of OBJBOX)
                                                             (IDIFFERENCE (PANEWIDTH PANE)
                                                                          X))
                                               HEIGHT _ (fetch YSIZE of OBJBOX))
                                DS)
                '(PROGN (DSPCLIPPINGREGION OLDVALUE ,DS])
```

(\TEDIT.OPERATE.OBJECT

```
  [LAMBDA (TSTREAM SEL PANE OPERATION)                              ; Edited 31-Dec-2024 17:24 by rmk
                                                                    ; Edited  1-Dec-2024 11:55 by rmk
                                                                    ; Edited 18-Oct-2024 13:44 by rmk
                                                                    ; Edited  6-Oct-2024 23:09 by rmk
                                                                    ; Edited 27-Aug-2024 10:03 by rmk
                                                                    ; Edited 21-Aug-2024 15:57 by rmk
                                                                    ; Edited 26-Jul-2024 14:50 by rmk
                                                                    ; Edited 20-Jul-2024 23:46 by rmk
```

```
    ;; SEL is a selection on a single image object in PANE, presumably where the mouse went down (SELPANE).  This executes that object's
    ;; WHENOPERATEDONFN.

    ;; If OPERATION is SELECTED, the function sees the true coordinate system of PANE, with PANE positioned at the hot-spot of the object in the
    ;; pane's coordinate system, simulating the setup in DISPLAYLINE..

    ;; Otherwise (the highlighting/showsel cases), the PANE's coordinate system has been restricted down to the object's:  0,0 is the lower-left corner,
    ;; everything outside of the object is clipped.

    ;; This difference is odd:  the WHENCHANGEDFN documentation is unclear about the set up for highlight vs. display. The WHENOPERATEDON
    ;; interface maybe wasn't thought out so well.

    ;; PANE presumably is the SELPANE.

    (LET* ((OBJ (FGETSEL SEL SELOBJ))
           (WHENOPERATEDONFN (IMAGEOBJPROP OBJ 'WHENOPERATEDONFN))
           (TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
           LINE)
          (CL:WHEN WHENOPERATEDONFN
              (SELECTQ OPERATION
                   (SELECTED ;; Called from BUTTONEVENTFN.DOOPERATION. Execute once, in PANE.  SHOWSEL and FIXSEL do the
                             ;; updates across other panes. This runs in PANE's coordinate system.  We can't do it if we can't determine from
                             ;; SEL where OBJ is located in PANE.
                             (CL:WHEN (SETQ LINE (\TEDIT.SEL.L1 SEL PANE TEXTOBJ))
                                 (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
                                 (MOVETO (FGETSEL SEL X0)
                                         (FGETLD LINE YBASE)
                                         PANE)
                                 (ERSETQ (APPLY* WHENOPERATEDONFN OBJ (WINDOWPROP PANE 'DSP)
                                                 OPERATION SEL TSTREAM))
                                 (\TEDIT.FIXSEL SEL TEXTOBJ)       ; Restore highlighting
                                 (\TEDIT.SHOWSEL SEL T TEXTOBJ)))
                   ((HIGHLIGHTED UNHIGHLIGHTED DESELECTED)
                                                ;; Execute in each pane where OBJ is visible, in OBJ's coordinate system.  This may be
                                                ;; duplicating the pane iteration in SHOWSEL?
                       [for L1 in (FGETSEL SEL L1) as P inpanes TEXTOBJ
                          when (AND L1 (INSIDE? (PANEREGION P)
                                                (FGETSEL SEL X0)
                                                (FGETLD L1 YBOT)))
                          do ;; The image object is visible in P's single line.
                             (RESETLST                          ; PROPAGATE THE ERROR?
                                 (\TEDIT.CLIP.OBJECT OBJ (FGETSEL SEL X0)
                                                     L1 P)
                                 (ERSETQ (APPLY* WHENOPERATEDONFN OBJ (WINDOWPROP PANE 'DSP)
                                                 OPERATION SEL TSTREAM)))])
                   (\TEDIT.THELP "BAD WHENOPERATEDON OPERATION" OPERATION)))])

)

;; SELPIECES

(DEFINEQ

(\TEDIT.SELPIECES
  [LAMBDA (SEL/FIRSTCHAR LASTCHAR TEXTOBJ)                  ; Edited 26-Nov-2024 17:49 by rmk
                                                            ; Edited 22-Nov-2024 14:24 by rmk
                                                            ; Edited  7-Jul-2024 09:10 by rmk
                                                            ; Edited 29-Apr-2024 13:13 by rmk
                                                            ; 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.

    ;;

    ;; Returns NIL if the arguments don't cover a valid piece sequence.

    ;;

    ;; 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 (if (FGETSEL SEL/FIRSTCHAR SET)
                     then (SETQ FIRSTCHAR (FGETSEL SEL/FIRSTCHAR CH#))
                          [SETQ LASTCHAR (CL:IF (EQ 0 (FGETSEL SEL/FIRSTCHAR DCH))
                                                FIRSTCHAR
                                                (SUB1 (FGETSEL SEL/FIRSTCHAR CHLIM)))]
                   else (SETQ FIRSTCHAR 0)
                        (SETQ LASTCHAR -1))
          elseif (type? TEDITHISTORYEVENT SEL/FIRSTCHAR)
            then (SETQ FIRSTCHAR (GETTH SEL/FIRSTCHAR THCH#))
                 (SETQ LASTCHAR (SUB1 (GETTH SEL/FIRSTCHAR THCHLIM)))
          else (SETQ FIRSTCHAR SEL/FIRSTCHAR))
        (CL:WHEN (AND (IBETWEENP FIRSTCHAR 1 (TEXTLEN TEXTOBJ))
                      (IBETWEENP LASTCHAR FIRSTCHAR (TEXTLEN TEXTOBJ)))
```

;; 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 26-Nov-2024 23:31 by rmk
                                                                       ; Edited 22-Nov-2024 15:38 by rmk
                                                                       ; 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:WHEN SELPIECES
    (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 (SETPC PREVPC NEXTPIECE NPC)
                                                                else (SETQ NEWFIRSTPIECE NPC))
                                                              (FSETPC NPC PREVPIECE 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  7-Nov-2024 21:50 by rmk
                                                                       ; Edited  4-Oct-2024 08:41 by rmk
```

; Edited 28-Apr-2024 08:52 by rmk
; 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 \TEDIT.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 [LET [(STR (ALLOCSTRING (PLEN PC]
                             ;; 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.
                             [for I from 1 to (PLEN PC) do (RPLCHARCODE STR I (APPLY* CHARFN
                                                                                      (\TEDIT.PIECE.NTHCHARCODE
                                                                                       TEXTOBJ PC I]
                             (if (fetch (STRINGP FATSTRINGP) of STR)
                                 then (FSETPC PC PTYPE FATSTRING.PTYPE)
                                      (FSETPC PC PBYTESPERCHAR 2)
                                      (FSETPC PC PBINABLE NIL)
                               else (FSETPC PC PTYPE THINSTRING.PTYPE)
                                    (FSETPC PC PBYTESPERCHAR 1)
                                    (FSETPC PC PBINABLE T))
                             (FSETPC PC PCONTENTS STR)
                             (FSETPC PC PBYTELEN (ITIMES (PBYTESPERCHAR PC)
                                                         (PLEN PC])
           (OBJECT.PTYPE (CL:WHEN OBJECTSTOO
                             (FSETPC PC PCONTENTS (APPLY* CHARFN PCONTENTS TEXTOBJ))))
           (SUBSTREAM.PTYPE
               (\TEDIT.THELP "SUBSTREAM PIECES NOT IMPLEMENTED"))
           (\TEDIT.THELP "ILLEGAL PIECE TYPE" (PTYPE PC]
   SELPIECES])
```

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

```
[LAMBDA (STRING TEXTOBJ CHECKFOREOL CHARLOOKS PARALOOKS)
```

; Edited 8-Feb-2025 20:14 by rmk
; 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 DEFAULTPARALOOKS)))
(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  7-Nov-2024 21:50 by rmk
                                                                ; Edited  4-Oct-2024 08:41 by rmk
                                                                ; 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
                                         (\TEDIT.THELP "SUBSTREAM PIECES NOT IMPLEMENTED"))
                                     (\TEDIT.THELP "ILLEGAL PIECE TYPE" (PTYPE PC)))
        finally (RETURN RESULT])

)

;; User entries to the selection code

(DEFINEQ
```

(TEDIT.XYTOCH

```
  [LAMBDA (X Y PANE)                                            ; Edited  6-Dec-2024 11:55 by rmk
                                                                ; Edited  1-Dec-2024 11:28 by rmk
                                                                ; Edited 29-Nov-2024 09:14 by rmk
                                                                ; Edited 20-Nov-2024 11:27 by rmk
                                                                ; Edited  8-Jul-2024 22:25 by rmk
                                                                ; Edited 28-Jun-2024 15:21 by rmk
                                                                ; Edited 25-Jun-2024 15:24 by rmk
                                                                ; Edited 20-Apr-2024 13:00 by rmk
                                                                ; Edited 20-Mar-2024 14:32 by rmk
```

;; Returns the character number of the character at coordinates X and Y in PANE. X and Y can be keywords LEFT/RIGHT/TOP/BOTTOM or

```
;; coordinates. Does not change the documents SEL.

;; Assumes that the keyword coordinates (as well as numeric X and Y) are relative to PANE's clipping region and not some subregion.

(LET ((SCRSEL (create SELECTION)))
      ;; 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 (PANELEFT PANE))
                   (RIGHT                                          ; Region RIGHT has the SUB1, but also adds LEFT.  Not sure
                          (SUB1 (PANEWIDTH PANE)))
                   X))
      (SETQ Y (SELECTQ Y
                   (TOP                                            ; Region TOP has the SUB1, but also adds BOTTOM.  Not sure
                        (SUB1 (PANEHEIGHT PANE)))
                   (BOTTOM (PANEBOTTOM PANE))
                   Y))
      (\TEDIT.XYTOSEL X Y SCRSEL (PANETOBJ PANE)
             PANE
             'NORMAL
             'LEFT NIL 'TEXT)
      (CL:WHEN (FGETSEL SCRSEL SET)
             (FGETSEL SCRSEL CH#])
```

(TEDIT.SELPROP

```
  [LAMBDA X                                                        ; Edited 28-Feb-2025 17:14 by rmk
                                                                   ; Edited  6-Feb-2025 16:48 by rmk
                                                                   ; Edited 31-Oct-2024 18:00 by rmk
                                                                   ; Edited 23-Sep-2024 23:11 by rmk
                                                                   ; Edited 22-Sep-2024 11:20 by rmk
                                                                   ; Edited 19-Aug-2024 13:55 by rmk
                                                                   ; Edited  7-Jul-2024 12:06 by rmk
                                                                   ; Edited 29-Apr-2024 08:31 by rmk
                                                                   ; Edited  6-Apr-2024 20:33 by rmk
                                                                   ; Edited  5-Apr-2024 12:29 by rmk
                                                                   ; Edited  2-Apr-2024 13:27 by rmk
    ;; User entry to get and set the properties of an external selection So that SELTEXTSTREAM is OK (A selection from TEDIT.GETSEL is a copy,
    ;; TEDIT.SETSEL is needed to install it.)

    (CL:UNLESS (IGEQ X 2)
           (\ILLEGAL.ARG X))
    (LET ([SEL (if (type? SELECTION (ARG X 1))
                   then (ARG X 1)
                 else (\DTEST (GETTOBJ (TEXTOBJ (ARG X 1))
                                       SEL)
                              'SELECTION]
          (PROP (ARG X 2))
          NEWVALUE)
         (PROG1 (SELECTQ PROP
                     (CH# (FGETSEL SEL CH#))
                     (CHLIM (FGETSEL SEL CHLIM))
                     ((LENGTH DCH)
                          (FGETSEL SEL DCH))
                     (POINT (FGETSEL SEL POINT))
                     ((KIND SELKIND)
                          (FGETSEL SEL SELKIND))
                     (CHLAST (if (EQ 0 (FGETSEL SEL DCH))
                                 then (FGETSEL SEL CH#)
                               else (FGETSEL SEL CHLAST)))
                     (POINTCH# (TEDIT.GETPOINT (FGETSEL SEL SELTEXTSTREAM)
                                               SEL))
                     (SELOBJ (FGETSEL SEL SELOBJ))
                     (TEXTSTREAM (FGETSEL SEL SELTEXTSTREAM))
                     (SHADE (FGETSEL SEL HOW))
                     (SHADEHEIGHT (FGETSEL SEL HOWHEIGHT))
                     (SET (FGETSEL SEL SET))
                     (\ILLEGAL.ARG PROP))
                (CL:WHEN (IGREATERP X 2)
                     (SETQ NEWVALUE (ARG X 3))
                     (SELECTQ PROP
                          (CH# (FSETSEL SEL CH# NEWVALUE))
                          ((LENGTH DCH)
                               (\TEDIT.UPDATE.SEL SEL NIL NEWVALUE))
                          (POINT (FSETSEL SEL POINT (OR [CAR (MEMB NEWVALUE '(LEFT RIGHT]
                                                                (\ILLEGAL.ARG NEWVALUE))))
                          ((KIND SELKIND)
                               (FSETSEL SEL SELKIND (OR [CAR (MEMB NEWVALUE '(CHAR WORD LINE PARA VOLATILE]
                                                             (\ILLEGAL.ARG NEWVALUE))))
                          (CHLAST (\TEDIT.UPDATE.SEL SEL NIL (IDIFFERENCE (ADD1 NEWVALUE)
                                                                          (FGETSEL SEL CH#))))
                          (CHLIM (\TEDIT.UPDATE.SEL SEL NIL (IDIFFERENCE NEWVALUE (FGETSEL SEL CH#))))
                          (SHADE (FSETSEL SEL HOW NEWVALUE))
                          (SHADEHEIGHT (FSETSEL SEL HOWHEIGHT NEWVALUE))
                          (SET (FSETSEL SEL SET NEWVALUE))
                          (\ILLEGAL.ARG PROP))
                (CL:WHEN (FGETSEL SEL SELTEXTSTREAM)
                     (\TEDIT.FIXSEL SEL (FGETSEL SEL SELTEXTSTREAM)))))])
```

(TEDIT.GETPOINT

```
[LAMBDA (TSTREAM SEL)                                    ; Edited 31-Oct-2024 17:46 by rmk
                                                         ; Edited 29-Oct-2024 11:47 by rmk
                                                         ; Edited  4-Oct-2024 08:41 by rmk
                                                         ; Edited 29-Apr-2024 10:49 by rmk
                                                         ; 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 TSTREAM's current
    ;; selection.  SEL can also be a character number, which is simply returned.

    (LET ((TEXTOBJ (TEXTOBJ TSTREAM)))
         (if (FIXP SEL)
             then (CL:UNLESS (AND (ILEQ SEL (ADD1 (TEXTLEN TEXTOBJ)))
                                  (IGEQ SEL 1))
                          (\ILLEGAL.ARG SEL))
                  SEL
           else (CL:UNLESS SEL
                   (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
                ;; LEFT and RIGHT are the same for a point (DCH=0) selection.
                (SELECTION! SEL)
                (SELECTQ (FGETSEL SEL POINT)
                     (LEFT (FGETSEL SEL CH#))
                     (RIGHT                              ; CHLIM is probaby not set appropriately for a RIGHT point
                                                         ; selection
                            (CL:IF (ZEROP (FGETSEL SEL DCH))
                                (ADD1 (FGETSEL SEL CH#))
                                (FGETSEL SEL CHLIM)))
                     (\TEDIT.THELP "Selection's POINT is neither RIGHT nor LEFT." (FGETSEL SEL POINT])
```

(TEDIT.GETSEL

```
[LAMBDA (TSTREAM)                                        ; Edited  7-Jul-2024 11:18 by rmk
                                                         ; Edited  1-May-2023 21:07 by rmk
                                                         ; Edited 30-May-91 23:03 by jds
    ;; This returns a copy of TSTREAM's current SEL for external use.  The textstream never points to this selection.  As long as this selection is alive,
    ;; it holds down the TSTREAM.

    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    (create SELECTION using (FGETTOBJ (TEXTOBJ TSTREAM)
                                      SEL)
                            SELTEXTSTREAM _ 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.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 18-May-2024 16:20 by rmk
                                                         ; Edited 29-Apr-2024 13:03 by rmk
                                                         ; Edited  9-Mar-2024 12:04 by rmk
                                                         ; Edited 15-Mar-2024 13:34 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))
      (TEXTOBJ (TEXTOBJ SEL)))
     (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
     (\TEDIT.SET.SEL.LOOKS SEL OPERATION)
     (\TEDIT.SHOWSEL SEL WASON TEXTOBJ)
     SEL])
```

(TEDIT.SETSEL

```
[LAMBDA (TSTREAM CH# LEN POINT PENDINGDELFLG LEAVECARETLOOKS OPERATION)
                                                                    ; Edited 17-Feb-2025 12:26 by rmk
                                                                    ; Edited 31-Jan-2025 12:43 by rmk
                                                                    ; Edited 19-Jan-2025 08:32 by rmk
                                                                    ; Edited  8-Jan-2025 00:20 by rmk
                                                                    ; Edited 26-Nov-2024 23:51 by rmk
                                                                    ; Edited 30-Jul-2024 23:27 by rmk
                                                                    ; Edited  7-Jul-2024 11:18 by rmk
                                                                    ; Edited 15-Jun-2024 10:08 by rmk
                                                                    ; Edited 23-May-2024 09:13 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 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.

;; For convenience, TSTREAM may be provided as an external selection (with its SELTEXTSTREAM as the actual TSTREAM). That selection is ;; never installed in TSTREAM, to avoid circularity.

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(CL:WHEN (AND LEN (ILESSP LEN 0))
       (ERROR "Selection length cannot be negative" LEN))
(LET* ((TEXTOBJ (TEXTOBJ! (GETTSTR TSTREAM TEXTOBJ)))
       (SEL (TEXTSEL TEXTOBJ))
       (TEXTLEN (TEXTLEN TEXTOBJ))
       PC)
      (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)                          ; First turn the old sel off.
      [if (type? SELECTION CH#)
          then                                                  ; He gave us a selection;  just plug it in
               (\TEDIT.COPYSEL CH# SEL)                         ; And make sure it can be turned on.
               (SETSEL SEL ONFLG NIL)
        else                                                    ; 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
             (CL:WHEN (ILESSP CH# 0)                            ; Negative => from end
                 (SETQ CH# (IPLUS 1 TEXTLEN CH#)))
             (if (EQ 0 TEXTLEN)
                 then (\TEDIT.UPDATE.SEL SEL 1 0 'LEFT)
               elseif (IGREATERP CH# TEXTLEN)
                 then (\TEDIT.UPDATE.SEL SEL TEXTLEN 0 'RIGHT)
               else [SETQ LEN (IMIN LEN (ADD1 (IDIFFERENCE TEXTLEN CH#]
                    (\TEDIT.UPDATE.SEL SEL CH# LEN POINT)
                    (FSETSEL SEL SELOBJ (CL:WHEN (EQ 1 LEN)
                                               (SETQ PC (\TEDIT.CHTOPC (GETSEL SEL CH#)
                                                                       TEXTOBJ))
                                               (CL:WHEN (EQ OBJECT.PTYPE (PTYPE PC))
                                                      (PCONTENTS PC)))]
      (SETTOBJ TEXTOBJ BLUEPENDINGDELETE PENDINGDELFLG)
      (\TEDIT.SET.SEL.LOOKS SEL OPERATION)
      (CL:UNLESS LEAVECARETLOOKS                                ; Set the insertion looks to follow.
          (SETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL)))
      (\TEDIT.FIXSEL SEL TEXTOBJ)
      (\TEDIT.SHOWSEL SEL T TEXTOBJ)
      (FSETTOBJ TEXTOBJ LASTARROWX NIL)
      (TEDIT.GETSEL TSTREAM])
```

(TEDIT.SHOWSEL

```
[LAMBDA (TSTREAM ONFLG SEL)                                         ; Edited  7-Jul-2024 11:25 by rmk
                                                                    ; Edited 18-May-2024 16:28 by rmk
                                                                    ; Edited 29-Apr-2024 12:27 by rmk
                                                                    ; Edited  9-Mar-2024 12:06 by rmk
                                                                    ; Edited 15-Mar-2024 13:36 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; Edited 30-May-91 23:04 by jds
```

;; He's giving us a selection to highlight and to connect it to this textobj.

```
(LET ((TEXTOBJ (TEXTOBJ TSTREAM)))
     (CL:UNLESS SEL
         (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
```

```
        (CL:WHEN SEL
            (\TEDIT.FIXSEL SEL TEXTOBJ)
            (\TEDIT.SHOWSEL SEL ONFLG TEXTOBJ))])
```

(TEDIT.SEL.AS.STRING

```
  [LAMBDA (TSTREAM SEL/CH# LEN CODEFOROBJECT)                    ; Edited 15-Feb-2025 12:47 by rmk
                                                                 ; Edited 14-Jul-2024 00:12 by rmk
                                                                 ; 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/CH#
        (SETQ SEL/CH# (GETTOBJ (GETTSTR TSTREAM TEXTOBJ)
                               SEL)))
    (LET (RESULT CH#)
         (if (type? SELECTION SEL/CH#)
             then (SETQ LEN (GETSEL SEL/CH# DCH))
                  (SETQ CH# (GETSEL SEL/CH# CH#))
           else (SETQ CH# SEL/CH#))
         (if (ZEROP LEN)
             then                                                ; There is no selection, or it's zero-width.  Return ''
                  (CONCAT "")
           else (SETQ RESULT (ALLOCSTRING LEN (CHARCODE SPACE)))  ; The resulting string
                (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 CH#))        ; Starting point for the string is start of selection.
                (for I C from 1 to LEN do (SETQ C (BIN TSTREAM))
                                          (CL:WHEN (AND (IMAGEOBJP C)
                                                        CODEFOROBJECT)
                                                                  ; RPLCHARCODE will cause an error on objects
                                              (SETQ C CODEFOROBJECT))
                                          (RPLCHARCODE RESULT I C))
            RESULT])
```

(TEDIT.SEL.AS.SEXPR

```
  [LAMBDA (TSTREAM SEL RDTBL FLG)                                ; Edited 29-Dec-2024 08:47 by rmk
                                                                 ; Edited 29-Apr-2024 10:49 by rmk
                                                                 ; 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 TSTREAM (TEDIT.GETPOINT TSTREAM SEL)
                                                            (TEDIT.ATOMBOUND.READTABLE (OR RDTBL *READTABLE*]
    (CAR (NLSETQ (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])

)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA TEDIT.SELPROP)
)
```

FUNCTION INDEX

MACRO INDEX

CONSTANT INDEX

RECORD INDEX

VARIABLE INDEX

I.S.OPR INDEX