File created: 16-Feb-2025 23:34:57 **{WMEDLEY}<library>tedit>TEDIT-WINDOW.;759**

edit by: rmk

changes to: (FNS \TEDIT.WINDOW.CREATE)

previous date: 13-Feb-2025 20:49:31 {WMEDLEY}<library>TEDIT>TEDIT-WINDOW.;754

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-WINDOWCOMS
       [(DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS TEDITCARET TEXTWINDOW PANEPROPS)
                                                 (MACROS FGETPANEPROP GETPANEPROP SETPANEPROP FSETPANEPROP)
                                                 (MACROS PANEPROPS PANEPREFIX PANESUFFIX PANETOPLINE PANECARET
                                                        PANESTREAM PANETOBJ PANEBOTTOMLINE \TEDIT.PREFIX.LCHARLIM
                                                        )
                                                 (MACROS PANETOP PANEWIDTH PANELEFT PANERIGHT PANEBOTTOM
                                                        PANEHEIGHT PANEREGION)
                                                 (I.S.OPRS inpanes backpanes)
                                                 (MACROS ALLBUTTONSUP)))
        (INITRECORDS TEDITCARET PANEPROPS)
        (FILES ATTACHEDWINDOW)
        (FNS TEDIT.DEFER.UPDATES)
        (FNS \TEDIT.WINDOW.CREATE \TEDIT.WINDOW.SETUP \TEDIT.MINIMAL.WINDOW.SETUP \TEDIT.CLEARPANE
             \TEDIT.FILL.PANES)
        (FNS \TEDIT.CURSORMOVEDFN \TEDIT.CURSOROUTFN \TEDIT.ACTIVE.WINDOWP \TEDIT.EXPANDFN \TEDIT.MAINW
             \TEDIT.MAINSTREAM \TEDIT.PRIMARYPANE \TEDIT.PANELIST \TEDIT.NEWREGIONFN \TEDIT.SET.WINDOW.EXTENT
             \TEDIT.SHRINK.ICONCREATE \TEDIT.SHRINKFN \TEDIT.PANEREGION)
        ;; Button events
        (FNS \TEDIT.BUTTONEVENTFN \TEDIT.BUTTONEVENTFN.DOOPERATION \TEDIT.BUTTONEVENTFN.GETOPERATION
             \TEDIT.BUTTONEVENTFN.CURSEL.INIT \TEDIT.BUTTONEVENTFN.INACTIVE \TEDIT.BUTTONEVENTFN.INTITLE
             \TEDIT.COPYINSERTFN \TEDIT.FOREIGN.COPY)
        (FNS \TEDIT.PANE.SPLIT \TEDIT.SPLITW \TEDIT.UNSPLITW \TEDIT.LINKPANES \TEDIT.UNLINKPANE)
        (P (MOVD? 'NILL 'GRAB-TYPED-REGION)
           (MOVD? 'NILL 'REGISTER-TYPED-REGION))
        (INITVARS (\TEDIT.OP.WIDTH 12)
               (\TEDIT.OP.BOTTOM 12)
               (\TEDIT.LINEREGION.WIDTH 12))
        (DECLARE%: DONTEVAL@LOAD DOCOPY (GLOBALVARS \TEDIT.OP.WIDTH \TEDIT.OP.BOTTOM \TEDIT.LINEREGION.WIDTH))
        (CURSORS BXCARET BXHICARET \TEDIT.LINECURSOR \TEDIT.SPLITCURSOR \TEDIT.MOVESPLITCURSOR
               \TEDIT.UNSPLITCURSOR \TEDIT.MAKESPLITCURSOR)
        (COMS                                                    ; User-level "is this a TEdit window?" function.
              (FNS TEDITWINDOWP))
        (COMS                                                    ; User-typein support
              (FNS TEDIT.GETINPUT \TEDIT.MAKEFILENAME))
        (COMS                                                    ; Attached Prompt window support.
              (FNS TEDIT.PROMPTWINDOW TEDIT.PROMPTPRINT TEDIT.PROMPTCLEAR TEDIT.PROMPTFLASH
                   \TEDIT.PROMPT.PAGEFULLFN)
              (INITVARS (TEDIT.PROMPT.FONT (FONTCREATE 'TERMINAL 10))
                     (TEDIT.PROMPTWINDOW.HEIGHT NIL))
              (GLOBALVARS TEDIT.PROMPT.FONT TEDIT.PROMPTWINDOW.HEIGHT))
        (COMS                                                    ; Title creation and update
              (FNS \TEXTSTREAM.TITLE \TEDIT.DEFAULT.TITLE \TEDIT.WINDOW.TITLE \TEXTSTREAM.FILENAME
                   \TEDIT.UPDATE.TITLE))
        (COMS                                                    ; Screen updating utilities
              (FNS TEDIT.DEACTIVATE.WINDOW \TEDIT.RESHAPEFN \TEDIT.REPAINTFN)
              (FNS \TEDIT.SCROLLFN \TEDIT.SCROLLCH.TOP \TEDIT.SCROLLCH.BOTTOM \TEDIT.SCROLLUP
                   \TEDIT.TOPLINE.YTOP \TEDIT.SCROLLDOWN \TEDIT.SCROLL.CARET \TEDIT.VISIBLECARETP
                   \TEDIT.VISIBLECHARP \TEDIT.BITMAPLINES \TEDIT.SETPANE.TOPLINE \TEDIT.SHIFTLINES)
              (FNS \TEDIT.ONSCREEN? \TEDIT.ONSCREEN.REGION \TEDIT.AFTERMOVEFN OFFSCREENP))
        (COMS                                                    ; Process-world interfaces
              (FNS \TEDIT.PROCIDLEFN \TEDIT.PROCENTRYFN \TEDIT.PROCEXITFN))
        (COMS (INITVARS (\CARETRATE 333))
                                                                 ; Caret handler;  stolen from CHAT.
              (FNS \TEDIT.DOWNCARET \TEDIT.FLASHCARET \TEDIT.UPCARET TEDIT.NORMALIZECARET \TEDIT.SETCARET
                   \TEDIT.CARET))
        [COMS                                                    ; Menu interfacing
              (FNS TEDIT.ADD.MENUITEM TEDIT.DEFAULT.MENUFN TEDIT.REMOVE.MENUITEM \TEDIT.CREATEMENU
                   \TEDIT.MENU.WHENHELDFN \TEDIT.MENU.WHENSELECTEDFN)
              (GLOBALVARS TEDIT.DEFAULT.MENU)
              [DECLARE%: DONTEVAL@LOAD DOCOPY (VARS (TEDIT.DEFAULT.MENU (\TEDIT.CREATEMENU
                                                                         '((Put 'Put NIL (SUBITEMS
                                                                                           |Put Formatted Document|
                                                                                                  Plain-Text))
                                                                           (Get 'Get NIL (SUBITEMS
                                                                                           |Get Formatted Document|
                                                                                                  Unformatted% Get
                                                                                                  ))
                                                                           Include Find Looks Substitute Quit
                                                                           (Expanded% Menu 'Expanded% Menu NIL
                                                                                  (SUBITEMS Expanded% Menu
                                                                                         Character% Looks
                                                                                         Paragraph% Formatting
```

```
                                                                                    Page% Layout]
            (DECLARE%: DONTEVAL@LOAD DOCOPY (P [OR (SASSOC 'TEdit BackgroundMenuCommands)
                                                   (NCONC1 BackgroundMenuCommands '(TEdit '(TEDIT)
                                                                                      "Opens a TEdit
                                                                                      window for use."]
                                                (SETQ BackgroundMenu NIL]
       (COMS                                                    ; titled icon info,
            (FILES ICONW)
            (BITMAPS TEDITICON TEDITMASK)
            (INITVARS (TEDIT.ICON.FONT (FONTCREATE 'HELVETICA 8 'BOLD))
                   (TEDIT.ICON.TITLE.REGION (CREATEREGION 16 4 64 77))
                   (TEDIT.TITLED.ICON.TEMPLATE (create TITLEDICON ICON _ TEDITICON MASK _ TEDITMASK TITLEREG _
                                                       TEDIT.ICON.TITLE.REGION])

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(DATATYPE TEDITCARET (TCNOWTIME

          (* Used to hold the current time, when checking to see if a transition is due)

                           TCTHENTIME                           (* Time when the next transition is to take place)
                           TCFORCEDDOWN                         (* TCFORCEDOWN = T means (Make the caret visible at the
                                                                next call to \EDIT.FLIPCARET.))
                           TCUP

          (* TCUP = T => The caret is NOT VISIBLE. Used to track the current state of the caret)

                           TCCARETDS                            (* The display stream that the caret appears in)
                           TCCURSORBM                           (* The CURSOR representing the caret)
                           TCCARETRATE                          (* %# of MSEC between caret up/down transitions)
                           TCFORCEUP

          (* T => The caret is not allowed to become visible. Used to keep the caret up during screen updates)

                           TCCARETX                             (* X position in the window that the caret appears at)
                           TCCARETY                             (* Y position in the window where the caret appears)
                           TCCARET                              (* A lisp CARET to be flashed (eventually))
                           )
       TCNOWTIME _ (CREATECELL \FIXP)
       TCTHENTIME _ (CREATECELL \FIXP)
       TCCURSORBM _ BXCARET TCCARETRATE _ \CARETRATE TCUP _ T TCCARET _ (\CARET.CREATE BXCARET))

[ACCESSFNS TEXTWINDOW ((WTEXTSTREAM (GETWINDOWPROP DATUM 'TEXTSTREAM)
                                    (PUTWINDOWPROP DATUM 'TEXTSTREAM NEWVALUE))
                       (WTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of (fetch (TEXTWINDOW WTEXTSTREAM) of DATUM)))
                       (PTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of (fetch (TEXTWINDOW WTEXTSTREAM) of DATUM)))
                       (CURSORREGION (GETWINDOWPROP DATUM 'TEDIT.CURSORREGION)
                                    (PUTWINDOWPROP DATUM 'TEDIT.CURSORREGION NEWVALUE))
                       (CLOSINGFILE (GETWINDOWPROP DATUM 'TEDIT-CLOSING-FILE)
                                    (PUTWINDOWPROP DATUM 'TEDIT-CLOSING-FILE NIL))
                       (PANEPROPS (GETWINDOWPROP DATUM 'PANEPROPS)
                                    (PUTWINDOWPROP DATUM 'PANEPROPS NEWVALUE)))
       (TYPE? (AND (WINDOWP DATUM)
                   (TYPENAMEP (fetch (TEXTWINDOW PTEXTOBJ) of DATUM)
                          'TEXTOBJ]

(DATATYPE PANEPROPS ((PWINDOW FULLXPOINTER)                     ; The window with these PANEPROPS
                     PREFIXLINE                                 ; Dummy line that covers all the characters above the first visible
                                                                ; line
                     SUFFIXLINE                                 ; Dummy line that covers all the characters below the last visible
                                                                ; line
                     PCARET NEXTPANE (PREVPANE XPOINTER)
                     PANEHEIGHT PANEWIDTH PANELEFT PANERIGHT PANEBOTTOM PANETOP PANEREGION))
)

(/DECLAREDATATYPE 'TEDITCARET '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                                        POINTER)
       ;; ---field descriptor list elided by lister---
       '22)

(/DECLAREDATATYPE 'PANEPROPS '(FULLXPOINTER POINTER POINTER POINTER POINTER XPOINTER POINTER POINTER POINTER
                                      POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '26)

(DECLARE%: EVAL@COMPILE

(PUTPROPS FGETPANEPROP MACRO ((P FIELD)
                              (ffetch (PANEPROPS FIELD) of P)))

(PUTPROPS GETPANEPROP MACRO ((P FIELD)
```

```
                                (fetch (PANEPROPS FIELD) of P)))

(PUTPROPS SETPANEPROP MACRO ((P FIELD NEWVALUE)
                             (replace (PANEPROPS FIELD) of P with NEWVALUE)))

(PUTPROPS FSETPANEPROP MACRO ((P FIELD NEWVALUE)
                              (freplace (PANEPROPS FIELD) of P with NEWVALUE)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS PANEPROPS MACRO ((PANE)
                           (fetch (TEXTWINDOW PANEPROPS) of PANE)))

(PUTPROPS PANEPREFIX MACRO ((PANE)
                            (LINEDESCRIPTOR! (GETPANEPROP (PANEPROPS PANE)
                                                          PREFIXLINE))))

(PUTPROPS PANESUFFIX MACRO ((PANE)
                            (GETPANEPROP (PANEPROPS PANE)
                                   SUFFIXLINE)))

(PUTPROPS PANETOPLINE MACRO ((PANE)
                             (FGETLD (PANEPREFIX PANE)
                                    NEXTLINE)))

(PUTPROPS PANECARET MACRO ((PANE)
                           (\DTEST (GETPANEPROP (PANEPROPS PANE)
                                                PCARET)
                                 'TEDITCARET)))

(PUTPROPS PANESTREAM MACRO ((PANE)
                            (fetch (TEXTWINDOW WTEXTSTREAM) of PANE)))

(PUTPROPS PANETOBJ MACRO [(PANE)
                          (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of (fetch (TEXTWINDOW WTEXTSTREAM) of PANE])

(PUTPROPS PANEBOTTOMLINE MACRO ((PANE)
                                (GETLD (PANESUFFIX PANE)
                                       PREVLINE)))

(PUTPROPS \TEDIT.PREFIX.LCHARLIM MACRO ((PANE CHNO)
                                        (FSETLD (PANEPREFIX PANE)
                                               LCHARLAST CHNO)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS PANETOP MACRO [(PANE PREG)
                         (fetch (REGION TOP) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANEWIDTH MACRO [(PANE PREG)
                           (fetch (REGION WIDTH) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANELEFT MACRO [(PANE PREG)
                          (fetch (REGION LEFT) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANERIGHT MACRO [(PANE PREG)
                           (fetch (REGION RIGHT) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANEBOTTOM MACRO [(PANE PREG)
                            (fetch (REGION BOTTOM) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANEHEIGHT MACRO [(PANE PREG)
                            (fetch (REGION HEIGHT) of (OR PREG (DSPCLIPPINGREGION NIL PANE])

(PUTPROPS PANEREGION MACRO ((PANE PREG)
                            (OR PREG (DSPCLIPPINGREGION NIL PANE))))
)

(DECLARE%: EVAL@COMPILE

[I.S.OPR 'inpanes NIL '(bind $$BODY _ BODY declare (LOCALVARS $$BODY)
                        first (SETQ I.V. (OR (CL:IF (TYPENAMEP $$BODY 'TEXTOBJ)
                                                    (FGETTOBJ $$BODY PRIMARYPANE)
                                                    $$BODY)
                                             (GO $$OUT)))
                        by (OR (GETPANEPROP (PANEPROPS I.V.)
                                            NEXTPANE)
                               (GO $$OUT]

[I.S.OPR 'backpanes NIL '(first (SETQ I.V. (OR (find P inpanes BODY suchthat (NULL (GETPANEPROP (PANEPROPS P)
                                                                                                 NEXTPANE)))
                                               (GO $$OUT)))
                          by (OR (GETPANEPROP (PANEPROPS I.V.)
                                              PREVPANE)
                                 (GO $$OUT]
```

```
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS ALLBUTTONSUP MACRO (NIL (ZEROP (LOGAND 7 LASTMOUSEBUTTONS))))
)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'TEDITCARET '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                                      POINTER)
        ;; ---field descriptor list elided by lister---
        '22)

(/DECLAREDATATYPE 'PANEPROPS '(FULLXPOINTER POINTER POINTER POINTER POINTER XPOINTER POINTER POINTER POINTER
                                    POINTER POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '26)

(FILESLOAD ATTACHEDWINDOW)

(DEFINEQ
```

(TEDIT.DEFER.UPDATES

```
  [LAMBDA (TSTREAM AFTERPROPS)                                          ; Edited 11-Jul-2024 12:26 by rmk
```

;; Suppresses selection/display updates (but not line layout) until a RESETLST is exited. At that point the display is updated (unless suppressed by ;; a higher deferral).

;; After props are installed after the display is updated.

;; This can be used to improve performance when a program is trying to construct an initial Tedit document, and also to avoid temporarily ;; inconsistent updates while a document is being modified.

```
    (LET ((TEXTOBJ (TEXTOBJ TSTREAM)))
         (RESETSAVE (TEXTPROP TEXTOBJ 'DON'TUPDATE T)
                `(PROGN (PUTTEXTPROP ,TEXTOBJ 'DON'TUPDATE OLDVALUE)
                        (\TEDIT.FILL.PANES ,TEXTOBJ)
                        (PUTTEXTPROPS ,TEXTOBJ ',AFTERPROPS])

)

(DEFINEQ
```

(\TEDIT.WINDOW.CREATE

```
  [LAMBDA (WINDOW TSTREAM PROPS)                                        ; Edited 16-Feb-2025 23:34 by rmk
                                                                        ; Edited  1-Jul-2024 22:55 by rmk
                                                                        ; Edited 29-Jun-2024 23:16 by rmk
                                                                        ; Edited  5-May-2024 21:54 by rmk
                                                                        ; Edited 20-Mar-2024 09:57 by rmk
                                                                        ; Edited 14-Jan-2024 22:13 by rmk
                                                                        ; Edited 18-Dec-2023 23:01 by rmk
                                                                        ; Edited 25-Nov-2023 10:37 by rmk
                                                                        ; Edited 23-Oct-2023 22:11 by rmk
                                                                        ; Edited 21-Oct-2023 12:20 by rmk
                                                                        ; Edited 18-Oct-2023 09:56 by rmk
                                                                        ; Edited  1-Jan-2022 23:54 by rmk
                                                                        ; Edited 30-Dec-2021 23:00 by rmk
                                                                        ; Edited 29-Dec-2021 16:35 by rmk
                                                                        ; Edited 24-Dec-2021 19:21 by rmk
                                                                        (* jds "23-May-85 15:19")
                                                                        ; Edited 27-Oct-2021 12:25 by rmk:
```

;; Don't set the global TEDIT default window if we have a REGIONTYPE, that must be special purpose.

;; If the region/window is typed, we grab (or create) a region of that type. The usual entry (TEDIT) defaults to type Tedit, giving a stack of regions in ;; TYPED-REGIONS. The effect is that the next (Tedit) window will open where the last Tedit window closed. It's a little tricky for ;; REGIONMANAGER to compensate for the prompt window, but it means that the user can reshape what is initially offered.

```
    (LET ((TEXTOBJ (TEXTOBJ TSTREAM))
          (PHEIGHT 0)
          TITLE REGIONTYPE PROMPTPROP REGION FILE PWINDOW PREPROMPT WTEXTOBJ WIDTH)
         (CL:WHEN (WINDOWP WINDOW)
             (CL:WHEN (SETQ WTEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of WINDOW))
                 ;; Reusing an existing Tedit window, undo its splits.
                 (for P backpanes WTEXTOBJ do (\TEDIT.UNSPLITW P)))
             [SETQ TITLE (OR (WINDOWPROP WINDOW 'TITLE)
                             (LISTGET PROPS 'TITLE])
         (SETQ REGIONTYPE (OR (GETTEXTPROP TEXTOBJ 'REGION-TYPE)
                              (AND (LITATOM WINDOW)
                                   WINDOW)))
         (SETQ FILE (GETTOBJ TEXTOBJ TXTFILE))
         (CL:UNLESS TITLE
             (SETQ TITLE (\TEDIT.DEFAULT.TITLE FILE PROPS)))
         (SETQ PROMPTPROP (GETTEXTPROP TEXTOBJ 'PROMPTWINDOW))
```

;; All this prompt-height calculation would be unnecessary if the attachment in GETPROMPTWINDOW does the proper shrinking of the main

```
        ;; window.
        (CL:UNLESS (EQ PROMPTPROP 'DONT'T)                          ; PHEIGHT remains 0 otherwise
            [SETQ PHEIGHT (if (WINDOWP PROMPTPROP)
                              then (SETQ PWINDOW PROMPTPROP)
                                   (FETCH (REGION HEIGHT) OF (WINDOWREGION PWINDOW 'REGION))
                            else (HEIGHTIFWINDOW (ITIMES (OR (GETTEXTPROP TEXTOBJ 'PROMPTWINDOWHEIGHT)
                                                             TEDIT.PROMPTWINDOW.HEIGHT 1)
                                                         (FONTPROP TEDIT.PROMPT.FONT 'HEIGHT])
        (CL:UNLESS (WINDOWP WINDOW)
            ;; If we can get an intended region first, we don't bother the user with a prompt
            (SETQ REGION (if (REGIONP WINDOW)
                             then (PROG1 (COPY WINDOW)
                                         (SETQ WINDOW NIL))
                           else (GRAB-TYPED-REGION REGIONTYPE)))
            (CL:UNLESS REGION
                (CLRPROMPT)                                         ; System promptwindow
                (printout PROMPTWINDOW "Please specify a " (OR REGIONTYPE "Tedit")
                        " window region")
                (CL:WHEN FILE
                    (printout PROMPTWINDOW " for " T "  " (FULLNAME FILE)))
                (TERPRI PROMPTWINDOW)
                [SETQ WIDTH (for PARALOOKS in (FGETTOBJ TEXTOBJ TXTPARALOOKSLIST) largest (GETPARA PARALOOKS
                                                                                           RIGHTMAR)
                                  finally (RETURN (IPLUS \TEDIT.LINEREGION.WIDTH (OR $$EXTREME 32)
                                                         12
                                                         (CL:IF (FGETTOBJ TEXTOBJ TXTNOTSPLITTABLE)
                                                             0
                                                             \TEDIT.OP.WIDTH)]
                (GETMOUSESTATE)
                [SETQ REGION (GETREGION 32 (IPLUS PHEIGHT 32)
                                  (CREATEREGION LASTMOUSEX LASTMOUSEY WIDTH (PLUS PHEIGHT 200]
                                                                    ; We don't want the default to keep shrinking
                (SETQ PREPROMPT (create REGION using REGION)))
            (add (fetch (REGION HEIGHT) of REGION)
                 (IMINUS PHEIGHT))
            (SETQ WINDOW (CREATEW REGION TITLE NIL NIL PROPS))
            ;; If we grabbed a typed-region, (maybe just a Tedit region by default.  We stash it back onto the window so it will be remembered for
            ;; next time.
            (REGISTER-TYPED-REGION REGION REGIONTYPE WINDOW))
        (WINDOWPROP WINDOW 'TEDITCREATED (OR PREPROMPT T))
        (CL:UNLESS [OR PWINDOW (EQ PROMPTPROP 'DON'T)
                       (SETQ PWINDOW (WINDOWP (CAR (WINDOWPROP WINDOW 'PROMPTWINDOW]
                                                                    ; Set up the promptwindow
            ;; GETPROMPTWINDOW sets WINDOW's PROMPTWINDOW to (PWINDOW . NLINES), but returns the window
            (SETQ PWINDOW (GETPROMPTWINDOW WINDOW (OR (GETTEXTPROP TEXTOBJ 'PROMPTWINDOWHEIGHT)
                                                      TEDIT.PROMPTWINDOW.HEIGHT 1)
                                  TEDIT.PROMPT.FONT)))
        (SETTOBJ TEXTOBJ PROMPTWINDOW PWINDOW)
        (CL:WHEN [WINDOWP (OR PWINDOW (SETQ PWINDOW (CAR (MKLIST PWINDOW]
            (WINDOWPROP PWINDOW 'PAGEFULLFN (FUNCTION \TEDIT.PROMPT.PAGEFULLFN))
            (WINDOWPROP PWINDOW 'TEDIT.PROMPTWINDOW T))
        ;; Make the window's dimensions available thru TSTREAM even though it hasn't yet been configured for the text
        (\TEDIT.MINIMAL.WINDOW.SETUP WINDOW TSTREAM PROPS)
        (WINDOWPROP WINDOW 'TITLE TITLE)
        WINDOW])
```

(\TEDIT.WINDOW.SETUP

```
  [LAMBDA (PANE TSTREAM PROPS AFTERPANE LCHAR1)                     ; Edited 25-Nov-2024 20:10 by rmk
                                                                    ; Edited 21-Nov-2024 21:12 by rmk
                                                                    ; Edited 18-Nov-2024 21:14 by rmk
                                                                    ; Edited  4-Nov-2024 19:47 by rmk
                                                                    ; Edited  3-Nov-2024 07:49 by rmk
                                                                    ; Edited  5-Jul-2024 11:38 by rmk
                                                                    ; Edited 18-May-2024 16:50 by rmk
                                                                    ; Edited 15-Mar-2024 13:36 by rmk
                                                                    ; Edited 29-Jan-2024 17:10 by rmk
                                                                    ; Edited 12-Oct-2023 23:41 by rmk
                                                                    ; Edited 10-May-2023 23:47 by rmk
                                                                    ; Edited  5-Nov-2022 23:13 by rmk
                                                                    ; Edited 11-Jun-99 15:48 by rmk:
                                                                    ; Edited 30-May-91 23:34 by jds
    ;; Set up PANE for display of TSTREAM's contents, treating PANE as a new (and possibly the only) pane. \TEDIT.MINIMAL.WINDOW.SETUP has
    ;; initialized PANE and installed it in its proper place.
    (CL:WHEN (EQ PANE AFTERPANE)
           (\TEDIT.THELP "PANE=AFTERPANE"))
    (\DTEST PANE 'WINDOW)
    (LET* ((TEXTOBJ (TEXTOBJ TSTREAM))
           (MENUPROP (LISTGET PROPS 'MENU))
           (SEL (TEXTSEL TEXTOBJ))
           LASTVISIBLE)                                             ; The Command menu, or list of items for it
          (if (type? MENU MENUPROP)
```

```
            then                                                    ; A menu.  just use it.
                (WINDOWPROP PANE 'TEDIT.MENU MENUPROP)
          elseif MENUPROP
            then                                                    ; Presumably a list of menu items.  Force a new menu on next
                                                                    ; middle button.
                (WINDOWPROP PANE 'TEDIT.MENU.COMMANDS MENUPROP)
                (WINDOWPROP PANE 'TEDIT.MENU NIL))

        ;;
        (\TEDIT.PANE.CREATELINES TEXTOBJ PANE (AND LCHAR1 (SUB1 LCHAR1)))
        (CL:UNLESS (OR LCHAR1 (EQ 0 (TEXTLEN TEXTOBJ)))
            (LINKLD (PANEPREFIX PANE)
                    (\TEDIT.FORMATLINE TEXTOBJ 1)))
        (CL:WHEN (PANETOPLINE PANE)
            [SETYBOT (PANEPREFIX PANE)
                    (IPLUS (FGETLD (PANETOPLINE PANE)
                                   LLEADBEFORE)
                           (fetch (REGION HEIGHT) of (DSPCLIPPINGREGION NIL PANE])
        (\TEDIT.CLEARPANE PANE)
        (\TEDIT.SUFFIXLINE.CREATE PANE TEXTOBJ (\TEDIT.LINES.BELOW NIL PANE TEXTOBJ))
        (CL:WHEN AFTERPANE
            (for PANE inpanes (PROGN TEXTOBJ) as L1 on (GETSEL SEL L1) as LN on (GETSEL SEL LN)
               when (EQ PANE AFTERPANE) do (push (CDR L1)
                                                 NIL)
                                           (push (CDR LN)
                                                 NIL)))
        (FSETSEL SEL HASCARET (NOT (FGETTOBJ TEXTOBJ TXTREADONLY)))
        (\TEDIT.FIXSEL SEL TEXTOBJ NIL (AND AFTERPANE PANE))
        (\TEDIT.SHOWSEL SEL NIL TEXTOBJ (AND AFTERPANE PANE))
        (\TEDIT.SHOWSEL SEL T TEXTOBJ (AND AFTERPANE PANE))
        (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ PANE])
```

(\TEDIT.MINIMAL.WINDOW.SETUP

```
  [LAMBDA (PANE TSTREAM PROPS AFTERPANE)                            ; Edited 30-Nov-2024 13:32 by rmk
                                                                    ; Edited  4-Nov-2024 19:46 by rmk
                                                                    ; Edited 26-Oct-2024 11:10 by rmk
                                                                    ; Edited 27-Aug-2024 10:11 by rmk
                                                                    ; Edited  6-Jul-2024 17:00 by rmk
                                                                    ; Edited  1-Jul-2024 09:20 by rmk
                                                                    ; Edited 30-Jun-2024 08:55 by rmk
                                                                    ; Edited 25-Jun-2024 00:04 by rmk
                                                                    ; Edited 13-Jun-2024 21:51 by rmk
                                                                    ; Edited 20-Mar-2024 11:22 by rmk
                                                                    ; Edited 22-Feb-2024 23:14 by rmk
                                                                    ; Edited 26-Jan-2024 13:14 by rmk
                                                                    ; Edited  2-Jan-2024 17:27 by rmk
                                                                    ; Edited 21-Dec-2023 17:19 by rmk
                                                                    ; Edited 20-Nov-2023 10:40 by rmk
                                                                    ; Edited  4-Oct-2023 09:48 by rmk
                                                                    ; Edited 18-Sep-2023 23:44 by rmk
                                                                    ; Edited 30-May-91 23:33 by jds
```

;; Do the minimum setup so that the window PANE becomes a pane of TSTREAM and TSTREAM and PANE know about each other. Does NOT ;; include mouse interface or scrolling/lines

;; If AFTERPANE is non-NIL, the new pnae will be placed after AFTERPANE in the TEXTOBJ's pane list. This maintains an ordering of panes, for ;; splitting and unsplitting.

```
        (\DTEST PANE 'WINDOW)
        (LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
              [PANEPROPS (create PANEPROPS
                                 PWINDOW _ PANE
                                 PCARET _ (create TEDITCARET
                                                  TCFORCEUP _ T
                                                  TCCARETDS _ (WINDOWPROP PANE 'DSP]
              DS PREG OLDPANES)                                     ; The displaystream for flashing the caret. Caret starts off, so it
                                                                    ; doesn't flash before its position is known
             (replace (TEXTWINDOW PANEPROPS) of PANE with PANEPROPS)
             (SETQ DS (WINDOWPROP PANE 'DSP))
             (FSETTOBJ TEXTOBJ SELPANE PANE)
             (WINDOWPROP PANE 'PROCESS NIL)                         ; For the moment, this pane has no process
             (replace (TEXTWINDOW WTEXTSTREAM) of PANE with TSTREAM)    ; TSTREAM is accessible from WINDOW
             (replace (TEXTWINDOW CURSORREGION) of PANE with (CREATEREGION 0 0 0 0))
                                                                    ; Used by CursorMovedFn
             (DSPRIGHTMARGIN 32767 DS)                              ; So we don't get spurious RETURNs printed out by the system
             (FSETTOBJ TEXTOBJ DISPLAYCACHE (CAR (\TEDIT.CREATE.LINECACHE 1)))
                                                                    ; A CACHE for creating line images for display
             [FSETTOBJ TEXTOBJ DISPLAYCACHEDS (DSPCREATE (fetch LCBITMAP of (GETTOBJ TEXTOBJ DISPLAYCACHE]
                                                                    ; A displaystream for changing the image caches
             (DSPOPERATION 'PAINT (FGETTOBJ TEXTOBJ DISPLAYCACHEDS))
             (DSPCLIPPINGREGION (create REGION
                                        LEFT _ 0
                                        BOTTOM _ 0
                                        WIDTH _ 100
                                        HEIGHT _ 15)
                    (FGETTOBJ TEXTOBJ DISPLAYCACHEDS))

        ;;
```

```
        (SETQ PREG (OR (LISTGET PROPS 'REGION)
                       (DSPCLIPPINGREGION NIL DS)))
        (WITH PANEPROPS PANEPROPS (SETQ PANEHEIGHT (fetch (REGION HEIGHT) of PREG))
              (SETQ PANEWIDTH (fetch (REGION WIDTH) of PREG))
              (SETQ PANELEFT (fetch (REGION LEFT) of PREG))
              (SETQ PANERIGHT (fetch (REGION RIGHT) of PREG))
              (SETQ PANEBOTTOM (fetch (REGION BOTTOM) of PREG))
              (SETQ PANETOP (fetch (REGION TOP) of PREG))
              (SETQ PANEREGION PREG))
        (WITH TEXTOBJ TEXTOBJ (SETQ WTOP (fetch (REGION PTOP) of PREG))
              (SETQ WRIGHT (fetch (REGION RIGHT) of PREG))
              (SETQ WBOTTOM (fetch (REGION BOTTOM) of PREG))
              (SETQ WLEFT (fetch (REGION LEFT) of PREG)))
       ;;
        (WINDOWPROP PANE 'CURSORMOVEDFN (FUNCTION \TEDIT.CURSORMOVEDFN))
        (WINDOWPROP PANE 'CURSOROUTFN (FUNCTION \TEDIT.CURSOROUTFN))
        (WINDOWPROP PANE 'BUTTONEVENTFN (FUNCTION \TEDIT.BUTTONEVENTFN))
        (WINDOWPROP PANE 'RIGHTBUTTONFN (FUNCTION \TEDIT.BUTTONEVENTFN))
        (WINDOWPROP PANE 'HARDCOPYFN (FUNCTION TEDIT.HARDCOPYFN))
        (WINDOWPROP PANE 'HARDCOPYFILEFN (FUNCTION \TEDIT.HARDCOPYFILEFN))
        (WINDOWPROP PANE 'COPYINSERTFN (FUNCTION \TEDIT.COPYINSERTFN))
        (WINDOWPROP PANE 'REPAINTFN (FUNCTION \TEDIT.REPAINTFN))
        (WINDOWPROP PANE 'AFTERMOVEFN (FUNCTION \TEDIT.AFTERMOVEFN))
        (WINDOWPROP PANE 'WINDOWENTRYFN (FUNCTION \TEDIT.PROCIDLEFN))
        (WINDOWPROP PANE 'OFFSCREEN (OFFSCREENP PANE))
        (WINDOWPROP PANE 'SCROLLFN (OR (WINDOWPROP PANE 'SCROLLFN)
                                       (FUNCTION \TEDIT.SCROLLFN)))
        (WINDOWPROP PANE 'ICONFN (OR (WINDOWPROP PANE 'ICONFN)
                                     (FUNCTION \TEDIT.SHRINK.ICONCREATE)))
        (WINDOWPROP PANE 'TEDIT.TITLEMENUFN (OR (WINDOWPROP PANE 'TEDIT.TITLEMENUFN)
                                                (LISTGET PROPS 'TITLEMENUFN)
                                                (FUNCTION TEDIT.DEFAULT.MENUFN)))
        (WINDOWADDPROP PANE 'SHRINKFN (FUNCTION \TEDIT.SHRINKFN))
        (WINDOWADDPROP PANE 'EXPANDFN (FUNCTION \TEDIT.EXPANDFN))
        (WINDOWADDPROP PANE 'RESHAPEFN (FUNCTION \TEDIT.RESHAPEFN))
        (WINDOWADDPROP PANE 'NEWREGIONFN (FUNCTION \TEDIT.NEWREGIONFN))
        ;; Our CLOSEFN must be first in order to stop closing if the stream is busy.
        (WINDOWADDPROP PANE 'CLOSEFN (CL:IF AFTERPANE
                                        [FUNCTION (LAMBDA (P)
                                                    (PUTWINDOWPROP P 'CLOSEFN NIL)
                                                    (\TEDIT.UNSPLITW P]
                                        (FUNCTION TEDIT.DEACTIVATE.WINDOW))
              T)
        (CL:UNLESS (thereis P inpanes TEXTOBJ suchthat (EQ P PANE))
                                                                ; Don't re-add
            (if AFTERPANE
                then                                            ; Link it in after AFTERPANE
                     (\TEDIT.LINKPANES AFTERPANE PANE)
              else (FSETTOBJ TEXTOBJ PRIMARYPANE PANE)))
        PANE])
```

(\TEDIT.CLEARPANE

```
  [LAMBDA (PANE PBOTTOM)                                        ; Edited  1-Dec-2024 11:43 by rmk
                                                                ; Edited 20-Nov-2024 11:31 by rmk
                                                                ; Edited  2-Jan-2024 11:13 by rmk
    ;; Clears PANE's clipping region.  PBOTTOM is usually NIL, but can focus clearing on a subregion.
    (CL:UNLESS PBOTTOM
        (SETQ PBOTTOM (PANEBOTTOM PANE)))
    (BLTSHADE WHITESHADE PANE 0 PBOTTOM (PANEWIDTH PANE)
           (IDIFFERENCE (fetch (REGION PTOP) of (PANEREGION PANE))
                  PBOTTOM)
           'REPLACE])
```

(\TEDIT.FILL.PANES

```
  [LAMBDA (TSTREAM ONLYPANE)                                    ; Edited 29-Nov-2024 13:29 by rmk
                                                                ; Edited 27-Nov-2024 13:51 by rmk
                                                                ; Edited 21-Nov-2024 21:10 by rmk
                                                                ; Edited 19-Nov-2024 23:27 by rmk
                                                                ; Edited 18-Nov-2024 21:14 by rmk
                                                                ; Edited 28-Oct-2024 16:29 by rmk
                                                                ; Edited 26-Oct-2024 15:38 by rmk
                                                                ; Edited  6-Jul-2024 16:57 by rmk
                                                                ; Edited 30-Jun-2024 17:12 by rmk
                                                                ; Edited 25-Jun-2024 08:53 by rmk
                                                                ; Edited 17-Jun-2024 09:36 by rmk
                                                                ; Edited 12-May-2024 21:36 by rmk
                                                                ; Edited 15-Mar-2024 13:36 by rmk
                                                                ; Edited 30-Nov-2023 10:02 by rmk
                                                                ; Edited 11-May-2023 11:35 by rmk
                                                                ; Edited 30-May-91 23:34 by jds
    ;; TSTREAM is a particular pane/window on calls for SHAPEW and REPAINT, but this refreshes that pane and all sister panes, in keeping with the
    ;; illusion that each pane is one part of a larger "window".
```

```
    ;; If called with a pane, the window system has cleared the bitmap, but we don't count on that.
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    (LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
          SEL WASON)
         (CL:WHEN TEXTOBJ
             (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
             (SETQ WASON (AND (GETSEL SEL SET)
                              (GETSEL SEL ONFLG)))
             (FSETSEL SEL ONFLG NIL)                          ; No highlighting for SEL to worry about: SEL and display will
                                                              ; both be off.
             (for P inpanes (PROGN TEXTOBJ) when (OR (NULL ONLYPANE)
                                                     (EQ P ONLYPANE))
                do ;; Take down the caret, and even clear PANE again, in case the timer had put it up again
                   (\TEDIT.SETCARET SEL P TEXTOBJ 'OFF)
                   (\TEDIT.CLEARPANE P)
                   (\TEDIT.SUFFIXLINE.CREATE P TEXTOBJ (\TEDIT.LINES.BELOW NIL P TEXTOBJ))
                   (\TEDIT.FIXSEL SEL TEXTOBJ NIL P)
                   (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ P))
             (CL:WHEN WASON (\TEDIT.SHOWSEL SEL T TEXTOBJ ONLYPANE)))])

)

(DEFINEQ

(\TEDIT.CURSORMOVEDFN
  [LAMBDA (PANE)                                              ; Edited  1-Dec-2024 11:55 by rmk
                                                              ; Edited 22-Nov-2024 23:53 by rmk
                                                              ; Edited 16-Nov-2024 20:18 by rmk
                                                              ; Edited 28-Jun-2024 15:07 by rmk
                                                              ; Edited 25-Jun-2024 14:53 by rmk
                                                              ; Edited 13-Jun-2024 22:29 by rmk
                                                              ; Edited 20-Mar-2024 11:00 by rmk
                                                              ; Edited 26-Jan-2024 12:48 by rmk
                                                              ; Edited  1-Oct-2022 16:07 by rmk
    ;; Watch the mouse and change the cursor to reflect the region of the pane it's in (line select, pane split eventually?)
    (PROG ((X (LASTMOUSEX PANE))
           (Y (LASTMOUSEY PANE))
           (TEXTOBJ (TEXTOBJ! (fetch (TEXTWINDOW WTEXTOBJ) of PANE)))
           (CURSORREG (fetch (TEXTWINDOW CURSORREGION) of PANE))
           LINE LEFT)
          (CL:UNLESS (INSIDE? (PANEREGION PANE)
                              X Y)
              (CURSOR T)
              (RETURN))
          (CL:UNLESS (INSIDE? CURSORREG X Y)
              [if (AND (IGEQ X (SETQ LEFT (IDIFFERENCE (FGETTOBJ TEXTOBJ WRIGHT)
                                                       \TEDIT.OP.WIDTH)))
                       (IGEQ Y (IPLUS (PANEBOTTOM PANE)
                                      \TEDIT.OP.BOTTOM))
                       (NOT (FGETTOBJ TEXTOBJ TXTNOTSPLITTABLE)))
                  then ;; We're in the split region on the right
                       (CURSOR \TEDIT.SPLITCURSOR)
                       (FSETTOBJ TEXTOBJ MOUSEREGION 'PANE)   ; PANE just signals \TEDIT.BUTTONEVENTFN to do a split
                                                              ; operation.
                       (replace (REGION LEFT) of CURSORREG with LEFT)
                       (replace (REGION WIDTH) of CURSORREG with \TEDIT.OP.WIDTH)
                  else ;; Not in the split region.  Are we in the line-select region on the left?  Don't call PANEPREFIX, because that tests for
                       ;; LINEDESCRIPTOR
                       (SETQ LINE (find L inlines (GETPANEPROP (PANEPROPS PANE)
                                                               PREFIXLINE)
                                          suchthat (ILEQ (FGETLD L YBOT)
                                                         Y)))
                       (CL:WHEN LINE                          ; The CURSORREGION picks out just LINE
                           (replace BOTTOM of CURSORREG with (FGETLD LINE YBOT))
                           (replace HEIGHT of CURSORREG with (FGETLD LINE LHEIGHT)))
                       ;; The line region gets wider if the paragraph is indented
                       (SETQ LEFT (OR (AND LINE (FGETLD LINE LEFTMARGIN))
                                      (IPLUS (FGETTOBJ TEXTOBJ WLEFT)
                                             \TEDIT.LINEREGION.WIDTH)))
                       (if (ILESSP X LEFT)
                           then ;; In left margin;  switch to the line-select cursor
                                (CURSOR \TEDIT.LINECURSOR)
                                (FSETTOBJ TEXTOBJ MOUSEREGION 'LINE)
                                (replace (REGION LEFT) of CURSORREG with 0)
                                (replace (REGION WIDTH) of CURSORREG with LEFT)
                         else ;; Not in the line-select region, not in the split region, must be the main text.
                              (CURSOR T)
                              (FSETTOBJ TEXTOBJ MOUSEREGION 'TEXT)
                              (replace (REGION LEFT) of CURSORREG with LEFT)
```

```
(replace (REGION WIDTH) of CURSORREG with (IDIFFERENCE (FGETTOBJ TEXTOBJ WRIGHT)
                                                        (IPLUS LEFT \TEDIT.LINEREGION.WIDTH])])
```

(\TEDIT.CURSOROUTFN

```
[LAMBDA (PANE)                                          ; Edited 20-Jul-2023 20:32 by rmk
                                                        ; Edited 30-May-91 23:32 by jds

    ;; Cursor leaves edit pane;  make sure we think we're in the text region.

    (CURSOR T)
    (SETTOBJ (fetch (TEXTWINDOW PTEXTOBJ) of PANE)
           MOUSEREGION
           'TEXT])
```

(\TEDIT.ACTIVE.WINDOWP

```
[LAMBDA (W)                                             ; Edited 20-Mar-2024 09:38 by rmk
                                                        ; Edited 15-Mar-2024 18:37 by rmk
                                                        ; Edited 11-Sep-2023 00:22 by rmk
                                                        ; Edited 30-May-91 23:33 by jds
    ;; RMK:  Not sure that TEXTOBJ is ever T. Or that windows ever have a TEXTSTREAM property (vs TEXTOBJ).

    ;; Decides whether a TEdit window is really in use.  The function TEDIT will set the TEXTOBJ prop of the window to T pro tem, to reserve a
    ;; window.  Once the TEdit has really started, the TEXTOBJ property will be a real textobj.

    (LET ((TEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of W)))
         (AND (type? TEXTOBJ TEXTOBJ)
              (NOT (fetch (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ))
              (PROCESSP (WINDOWPROP W 'PROCESS])
```

(\TEDIT.EXPANDFN

```
[LAMBDA (W)                                             (* jds " 7-May-85 15:56")
                                                        (* steals back the tty for us when the TEdit window is expanded.)

    (COND
       ((WINDOWPROP W 'PROCESS)                         (* There's a process to go with this edit window.
                                                        Give it the TTY.)

        (TTY.PROCESS (WINDOWPROP W 'PROCESS])
```

(\TEDIT.MAINW

```
[LAMBDA (TSTREAM)                                       ; Edited 20-Oct-2024 09:18 by rmk
                                                        ; Edited 26-Aug-2024 12:16 by rmk
                                                        ; Edited 25-Aug-2024 08:55 by rmk
                                                        ; Edited 28-Jun-2024 22:26 by rmk
                                                        ; Edited 19-Sep-2023 08:41 by rmk
                                                        ; Edited 11-Sep-2023 09:37 by rmk
                                                        ; Edited  6-May-2023 17:29 by rmk
                                                        ; Edited  5-Nov-2022 12:21 by rmk
                                                        ; Edited 30-May-91 23:33 by jds
    ;; If TSTREAM's window is an attached window whose main window is also a textstream, returns that main window.  Most likely TSTREAM is a
    ;; menu.

    (LET ((PRIM (\TEDIT.PRIMARYPANE TSTREAM)))
         (OR (AND PRIM (WINDOWPROP PRIM 'MAINWINDOW))
             PRIM])
```

(\TEDIT.MAINSTREAM

```
[LAMBDA (TSTREAM)                                       ; Edited 20-Oct-2024 09:31 by rmk
  (LET ((MAINW (\TEDIT.MAINW TSTREAM)))
       (CL:WHEN MAINW
           (fetch (TEXTWINDOW WTEXTSTREAM) of MAINW))])
```

(\TEDIT.PRIMARYPANE

```
[LAMBDA (TSTREAM)                                       ; Edited 28-Jun-2024 21:36 by rmk
                                                        ; Edited 25-Jun-2024 11:57 by rmk
                                                        ; Edited 19-Sep-2023 08:21 by rmk
                                                        ; Edited 30-May-91 23:33 by jds
    ;; This returns the first pane in the sequence of panes associated with TSTREAM, the original pane, before any splitting.  Note that this is different
    ;; than \TEDIT.MAINW: that maps from attached windows (e.g. menus) back to the primary pane.

    (GETTOBJ (TEXTOBJ TSTREAM)
           PRIMARYPANE])
```

(\TEDIT.PANELIST

```
[LAMBDA (TSTREAM/PANE)                                  ; Edited 30-Jun-2024 22:36 by rmk
    ;; Returns the panes as a list of windows, primarily so that \MODERNIZED.TEDIT.BUTTONEVENTFN can get the region without knowing about
    ;; the internal PANEPROPS record.

    (for P inpanes (if (type? TEXTWINDOW TSTREAM/PANE)
                       then TSTREAM/PANE
                     else (TEXTOBJ TSTREAM/PANE))
       collect P])
```

(\TEDIT.NEWREGIONFN

```
[LAMBDA (FIXEDPOINT MOVINGPOINT WINDOW)                    (* jds "24-FEB-83 17:43")

        (* This function is called whenever a new region for the window is needed.
        It constrains the size of the window so that the menu and/or titles will fit)

    (COND
       ((NULL MOVINGPOINT)                                 (* This is true only the first time the function is called)
        FIXEDPOINT)
       (T (PROG (%#OFMENUITEMS MENUWIDTH XDELTA YDELTA)

            (* The NEWREGIONFNARG can be either a window or a list consisting of the number of items in the menu and the
            minimum width of the window neede to hold the menu an titles)

                  (SETQ XDELTA (IDIFFERENCE (fetch (POSITION XCOORD) of MOVINGPOINT)
                                            (fetch (POSITION XCOORD) of FIXEDPOINT)))
                  (SETQ YDELTA (IDIFFERENCE (fetch (POSITION YCOORD) of MOVINGPOINT)
                                            (fetch (POSITION YCOORD) of FIXEDPOINT)))
                  [COND
                     [(IGEQ XDELTA 0)
                      (replace (POSITION XCOORD) of MOVINGPOINT with (IPLUS (fetch (POSITION XCOORD) of FIXEDPOINT)
                                                                            (IMAX 32 XDELTA]
                     (T (replace (POSITION XCOORD) of MOVINGPOINT with (IPLUS (fetch (POSITION XCOORD) of FIXEDPOINT)
                                                                              (IMIN -32 XDELTA]
                  [COND
                     [(IGEQ YDELTA 0)
                      (replace (POSITION YCOORD) of MOVINGPOINT with (IPLUS (fetch (POSITION YCOORD) of FIXEDPOINT)
                                                                            (IMAX 32 YDELTA]
                     (T (replace (POSITION YCOORD) of MOVINGPOINT with (IPLUS (fetch (POSITION YCOORD) of FIXEDPOINT)
                                                                              (IMIN -32 YDELTA]
                  (RETURN MOVINGPOINT])
```

(\TEDIT.SET.WINDOW.EXTENT

```
[LAMBDA (TEXTOBJ PANE)                                     ; Edited  1-Dec-2024 11:28 by rmk
                                                           ; Edited 29-Nov-2024 10:59 by rmk
                                                           ; Edited 17-Nov-2024 18:59 by rmk
                                                           ; Edited 28-Jun-2024 15:11 by rmk
                                                           ; Edited 13-Jun-2024 22:38 by rmk
                                                           ; Edited 11-Jan-2024 19:29 by rmk
                                                           ; Edited 20-Nov-2023 11:09 by rmk
                                                           ; Edited  3-Nov-2023 12:09 by rmk
                                                           ; Edited 22-Sep-2023 19:57 by rmk
                                                           ; Edited 11-May-2023 00:35 by rmk
                                                           ; Edited  4-May-2023 21:52 by rmk
                                                           ; Edited 28-Apr-2023 11:23 by rmk
                                                           ; Edited 15-Feb-2023 23:41 by rmk
                                                           ; Edited  3-Nov-2022 23:23 by rmk
                                                           ; Edited 30-May-91 23:33 by jds
    ;; Set the window's EXTENT property according to 1st and last char on screen.
    (CL:UNLESS (GETTEXTPROP TEXTOBJ 'NOEXTENT)
        (CL:WHEN PANE
            (LET ((TEXTLEN (FGETTOBJ TEXTOBJ TEXTLEN))
                  (PHEIGHT (PANEHEIGHT PANE))
                  (PBOTTOM (PANEBOTTOM PANE))
                  FIRSTLINE LASTLINE TOPCHAR BOTCHAR EXTHEIGHT EXTBOT YBOT)
                  ;; First visible line
                  (SETQ FIRSTLINE (find L inlines (PANEPREFIX PANE) suchthat (ILESSP (FGETLD L YBOT)
                                                                                     PHEIGHT)))
                  ;; Last visible line
                  (for L inlines FIRSTLINE while (IGEQ (FGETLD L YBOT)
                                                       PBOTTOM)
                     do (SETQ LASTLINE L))
                  ;; Start of first visible line
                  (SETQ TOPCHAR (CL:IF FIRSTLINE
                                       (FGETLD FIRSTLINE LCHAR1)
                                       TEXTLEN))
                  (COND
                     (LASTLINE
                            ;; There IS a last line on the screen.  Grab its last character as the bottom character on the screen, and set the
                            ;; lowest-Y position to the bottom of that line
                            (SETQ BOTCHAR (IMIN TEXTLEN (FGETLD LASTLINE LCHARLAST)))
                            (SETQ YBOT (FGETLD LASTLINE YBOT)))
                     (T ;; Everything is off the top of the screen.  Bottom character is also the last char in the document, and the lowest Y we
                        ;; encountered is the top of the edit window.
                        (SETQ BOTCHAR TEXTLEN)
                        (SETQ YBOT PHEIGHT)))
                  [COND
                     ((AND (IEQP BOTCHAR TEXTLEN)
                           (IEQP TOPCHAR TEXTLEN))                ; At the bottom of the document
                      (SETQ EXTBOT (SUB1 YBOT))
```

```
                    (SETQ EXTHEIGHT PHEIGHT))
                   (T ;; Otherwise, set the bottom in proportion to what is left below the bottom of the screen, and the extent height in
                      ;; proportion to how much text appears in the window
                      [SETQ EXTHEIGHT (FIXR (FQUOTIENT (ITIMES (IDIFFERENCE PHEIGHT YBOT)
                                                               TEXTLEN)
                                                        (IMAX (IDIFFERENCE BOTCHAR TOPCHAR)
                                                              1]
                      (SETQ EXTBOT (IDIFFERENCE YBOT (FIXR (FQUOTIENT (ITIMES (IDIFFERENCE PHEIGHT YBOT)
                                                                              (IDIFFERENCE TEXTLEN BOTCHAR))
                                                                      (IMAX (IDIFFERENCE BOTCHAR TOPCHAR)
                                                                            1]
                (WINDOWPROP PANE 'EXTENT (create REGION
                                                 BOTTOM _ EXTBOT
                                                 HEIGHT _ (IMAX 1 EXTHEIGHT)
                                                 WIDTH _ (PANEWIDTH PANE)
                                                 LEFT _ 0))))))])
```

(**TEDIT.SHRINK.ICONCREATE**

```
[LAMBDA (W ICON ICON-POSITION)                                  ; Edited 15-Mar-2024 18:28 by rmk
                                                                ; Edited 20-Dec-2023 23:44 by rmk
                                                                ; Edited 10-Apr-2023 09:44 by rmk
                                                                ; Edited 25-Apr-88 23:53 by jds
    ;; Create the icon that represents this window.
    [PROG [(ICON (WINDOWPROP W 'ICON))
           (ICONTITLE (WINDOWPROP W 'TEDIT.ICON.TITLE))
           (SHRINKFN (WINDOWPROP W 'SHRINKFN]
          (COND
             ((NOT (fetch (TEXTWINDOW WTEXTOBJ) of W))          ; This isn't really a TEdit window any more.  Don't do anything
              NIL)
             ((TEDITMENUP W)                                    ; This is a text menu, and shrinks without trace.
              NIL)
             ((OR (IGREATERP (FLENGTH SHRINKFN)
                             3)
                  (AND (NOT (FMEMB 'SHRINKATTACHEDWINDOWS SHRINKFN))
                       (IGREATERP (FLENGTH SHRINKFN)
                                  2)))                          ; There are other functions that expect to handle this.  Don't
                                                                ; bother.
              NIL)
             ((OR [AND ICONTITLE (EQUAL ICONTITLE (\TEXTSTREAM.TITLE (TEXTSTREAM W]
                  (AND (NOT ICONTITLE)
                       ICON))
              ;; we built this and the title is the same, or he has already put an icon on this.  Do nothing
              NIL)
             (ICON ;; There's an existing icon window;  change the title in it
                   [WINDOWPROP W 'TEDIT.ICON.TITLE (SETQ ICONTITLE (\TEXTSTREAM.TITLE (TEXTSTREAM W]
                   (ICONTITLE ICONTITLE NIL NIL ICON))
             (T                                                 ; install a new icon
                [WINDOWPROP W 'TEDIT.ICON.TITLE (SETQ ICONTITLE (\TEXTSTREAM.TITLE (TEXTSTREAM W]
                (WINDOWPROP W 'ICON (TITLEDICONW TEDIT.TITLED.ICON.TEMPLATE ICONTITLE TEDIT.ICON.FONT
                                                 ICON-POSITION T NIL 'FILE]
    (WINDOWPROP W 'ICON])
```

(**TEDIT.SHRINKFN**

```
[LAMBDA (W ICON ICONW)                                          ; Edited 24-Sep-2023 23:32 by rmk
                                                                (* jds "14-Dec-84 08:56")
    ;; Hands to othe EXEC process, or MOUSE if EXEC isn't found.
    (CL:WHEN (AND (EQ (WINDOWPROP W 'PROCESS)
                      (TTY.PROCESS)))
             (TTY.PROCESS T])
```

(**TEDIT.PANEREGION**

```
[LAMBDA (PANE)                                                  ; Edited  1-Dec-2024 11:44 by rmk
                                                                ; Edited 10-May-2023 23:15 by rmk
    ;; Value may be a shrunken version of PANE's clipping region, reduced to the subregion that is visible on the screen in its original coordinates.
    ;; That is, if the bottom is now 100 points below the screen, then 100 is added to BOTTOM and taken away from HEIGHT.
    (NOTUSED)
    (LET [(PREG (DSPCLIPPINGREGION NIL PANE))
          (WREG (WINDOWPROP PANE 'REGION]
         (if (OR (ILESSP (fetch (REGION LEFT) of WREG)
                         0)
                 (ILESSP (fetch (REGION BOTTOM) of WREG)
                         0)
                 (IGREATERP (fetch (REGION PRIGHT) of WREG)
                            SCREENWIDTH)
                 (IGREATERP (fetch (REGION PTOP) of WREG)
                            SCREENHEIGHT))
             then [LET [[LDIFF (IMAX 0 (IDIFFERENCE 0 (fetch (REGION LEFT) of WREG]
                        [BDIFF (IMAX 0 (IDIFFERENCE 0 (fetch (REGION BOTTOM) of WREG]
```

```
                        (RDIFF (IMAX 0 (IDIFFERENCE (fetch (REGION RIGHT) of WREG)
                                                    SCREENWIDTH)))
                        (TDIFF (IMAX 0 (IDIFFERENCE (fetch (REGION HEIGHT) of WREG)
                                                    SCREENHEIGHT]
                    ;; The diffs are positive or 0--how much is outside the screen and needs to be added/subtracted.
                    (CREATEREGION (IPLUS (fetch (REGION LEFT) of PREG)
                                         LDIFF)
                                  (IPLUS (fetch (REGION BOTTOM) of PREG)
                                         BDIFF)
                                  (IDIFFERENCE (fetch (REGION WIDTH) of PREG)
                                               (IPLUS LDIFF RDIFF))
                                  (IDIFFERENCE (fetch (REGION HEIGHT) of PREG)
                                               (IPLUS BDIFF TDIFF]
          else PREG])

)

;; Button events

(DEFINEQ

(\TEDIT.BUTTONEVENTFN
  [LAMBDA (PANE)                                                    ; Edited 13-Feb-2025 11:53 by rmk
                                                                    ; Edited  6-Dec-2024 11:33 by rmk
                                                                    ; Edited  1-Dec-2024 12:03 by rmk
                                                                    ; Edited 27-Nov-2024 20:21 by rmk
                                                                    ; Edited  3-Nov-2024 07:19 by rmk
                                                                    ; Edited 21-Oct-2024 00:18 by rmk
                                                                    ; Edited 19-Oct-2024 23:16 by rmk
                                                                    ; Edited  6-Oct-2024 20:48 by rmk
                                                                    ; Edited 24-Sep-2024 23:06 by rmk
                                                                    ; Edited 12-Sep-2024 17:51 by rmk
                                                                    ; Edited 20-Jul-2024 15:23 by rmk
                                                                    ; Edited 28-Jun-2024 22:52 by rmk
                                                                    ; Edited 25-May-2024 13:49 by rmk
                                                                    ; Edited 29-Apr-2024 13:43 by rmk
                                                                    ; Edited 24-Feb-2024 15:29 by rmk
                                                                    ; Edited 20-Jul-2023 21:52 by rmk
                                                                    ; Edited  9-Apr-2023 22:59 by rmk
                                                                    ; Edited 19-Sep-2021 22:58 by rmk:
    ;; Handle mouse buttons that are clicked in a TEdit pane.
    (TOTOPW PANE)
    (CL:WHEN (MOUSESTATE (OR LEFT MIDDLE RIGHT))
        ;; If no button is down, we got control on button-up transition, so ignore it.
        (RESETLST                                                   ; Getting TTYPROC here allows HELP in debugging
            (bind (TTYPROC _ (TTY.PROCESS))
                  (TSTREAM _ (PANESTREAM PANE))
                  (X _ (LASTMOUSEX PANE))
                  (Y _ (LASTMOUSEY PANE))
                  (DS _ (WINDOWPROP PANE 'DSP))
                  (OLDX _ MIN.SMALLP)
                  (OLDY _ MIN.SMALLP)
                  (PREG _ (PANEREGION PANE))
                  TEXTOBJ CURSEL NEWSEL CUROPERATION NEWOPERATION PENDINGDEL READONLY declare (SPECVARS CURSEL)
              first ;; Pick off and return from a bunch of peripheral situations, then fall through to the complexities of normal text selection.
                    (CL:UNLESS TSTREAM (RETURN))
                    (SETQ TEXTOBJ (TEXTOBJ! (FGETTSTR TSTREAM TEXTOBJ)))
                    (CL:WHEN (OR (\TEDIT.BUTTONEVENTFN.INTITLE Y PANE TEXTOBJ)
                                 (\TEDIT.BUTTONEVENTFN.INACTIVE TEXTOBJ PANE)
                                 (\TEDIT.PANE.SPLIT TEXTOBJ PANE))
                             (RETURN))
                    ;;
                    ;; The usual case -- a valid click in this pane.  And there's nothing else going on now.
                    ;; Make sure the caret isn't being displayed, then change to the special tall one so it is easier to see during typein
                    (\CARET.DOWN)
                    [RESETSAVE (\TEDIT.CARET TEXTOBJ BXHICARET)
                           `(\TEDIT.CARET ,TEXTOBJ ,BXCARET]
                    ;;
                    (SETQ READONLY (FGETTOBJ TEXTOBJ TXTREADONLY))
                    (SETQ CUROPERATION 'NORMAL)
                    (SETQ NEWOPERATION (\TEDIT.BUTTONEVENTFN.GETOPERATION READONLY NIL))
                    (CL:UNLESS (SETQ CURSEL (\TEDIT.BUTTONEVENTFN.CURSEL.INIT NEWOPERATION TEXTOBJ))
                           (RETURN))
                    (SETQ NEWSEL (\TEDIT.COPYSEL CURSEL))           ; Gets line-chains and consistent initial looks
                    (FSETTOBJ TEXTOBJ LASTARROWX NIL)
              eachtime (BLOCK)                                      ; Give other processes a chance
                       (GETMOUSESTATE)                              ; And get the new mouse and key info
                       (\TEDIT.CURSORMOVEDFN PANE)
                       (SETQ NEWOPERATION (\TEDIT.BUTTONEVENTFN.GETOPERATION READONLY CUROPERATION))
```

```
        ;; We're done if keys and buttons are up
until (AND (EQ NEWOPERATION 'NORMAL)
           (ALLBUTTONSUP))
unless (AND (IEQP OLDX (SETQ X (LASTMOUSEX DS)))
            (IEQP OLDY (SETQ Y (LASTMOUSEY DS)))
            (EQ CUROPERATION NEWOPERATION))
do  ;; Polling loop, track the mouse until the buttons and modifier keys come up, i.e. NORMAL Nothing to do until the mouse moves
    ;; or the operation changes. .

    ;; First and always: CURSEL is ON at this point and matches the display. NEWSEL may not be well-defined.

    (CL:UNLESS (INSIDEP (PANEREGION PANE PREG)
                        X Y)                          ; The mouse left the window: cleanup and leave.
        (CL:UNLESS (EQ CUROPERATION 'NORMAL)          ; Take down the copy/delete/copylooks highlight
            (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
            (\TEDIT.SHOWSEL NIL T TEXTOBJ))           ; Go back to original selection?

        ;; Scroll if mouse moved to scroll bar (and scroll bar doesn't overlap the window)

        (CL:WHEN (IN/SCROLL/BAR? PANE LASTMOUSEX LASTMOUSEY)
               (SCROLL.HANDLER PANE))
        (RETURN))

    ;;

    ;; Ready to track the selection.

    (SETQ OLDX X)
    (SETQ OLDY Y)
    (CL:UNLESS (EQ NEWOPERATION CUROPERATION)         ; Keys changed
        (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)           ; Switch to new highlighting
        (\TEDIT.SET.SEL.LOOKS CURSEL NEWOPERATION)
        (\TEDIT.SET.SEL.LOOKS NEWSEL NEWOPERATION)
        (CL:WHEN (EQ NEWOPERATION 'NORMAL)

            ;; Switching from e.g. COPY to NORMAL with button down. Since we didn't start out NORMAL, the original normal
            ;; selection is still on the screen.  We take it down here, and establish the current (off) CURSEL as the new
            ;; restoration selection

            (\TEDIT.SHOWSEL NIL NIL TEXTOBJ)
            (\TEDIT.COPYSEL CURSEL (TEXTSEL TEXTOBJ)))
        (\TEDIT.SHOWSEL CURSEL T TEXTOBJ)
        (SETQ CUROPERATION NEWOPERATION))

    ;; Update NEWSEL each time around. Note that \TEDIT.XYTOSEL fixes but doesn't show the selection, we do that here.
    ;; MOUSEREGION is set by \TEDITCURSORMOVEDFN, below.

    (if (\TEDIT.MOUSESTATE RIGHT)
        then                                          ; Right button: NEWSEL extends last CURSEL
            (\TEDIT.XYTOSEL X Y NEWSEL TEXTOBJ CUROPERATION PANE 'RIGHT CURSEL)
            (CL:WHEN (FGETSEL NEWSEL SET)
                (CL:WHEN (AND TEDIT.EXTEND.PENDING.DELETE (NOT PENDINGDEL)
                              (EQ CUROPERATION 'NORMAL)
                              (NOT (FGETTOBJ TEXTOBJ TXTREADONLY)))

                    ;; Switch to simulation of Laurel bluependingdelete: Black, deletes on type-in. Coerce CURSEL and
                    ;; display for pending looks.  Otherwise, CURSEL is already BPD and stays on to avoid flicker in
                    ;; extending

                    (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                                                      ; Take down old looks, change, re-show
                    (\TEDIT.SET.SEL.LOOKS CURSEL 'PENDINGDEL)
                    (\TEDIT.SET.SEL.LOOKS NEWSEL 'PENDINGDEL)
                    (\TEDIT.SHOWSEL CURSEL T TEXTOBJ)
                    (SETQ PENDINGDEL T))
                [\TEDIT.EXTEND.SEL NEWSEL CURSEL TEXTOBJ (MEMB CUROPERATION '(COPY COPYLOOKS])
                                                      ; No valid selection, go to cleanup

      else (if (\TEDIT.MOUSESTATE LEFT)
               then                                   ; Left selects char/point.
                   (\TEDIT.XYTOSEL X Y NEWSEL TEXTOBJ CUROPERATION PANE 'LEFT CURSEL)
             elseif (\TEDIT.MOUSESTATE MIDDLE)
               then                                   ; Middle selects word/line
                   (\TEDIT.XYTOSEL X Y NEWSEL TEXTOBJ CUROPERATION PANE 'MIDDLE CURSEL))
           (CL:WHEN (AND (FGETSEL NEWSEL SET)
                         (\TEDIT.SEL.CHANGED? NEWSEL CURSEL)
                         (OR (NOT (ALLBUTTONSUP))
                             (FGETSEL NEWSEL SELOBJ)))

                ;; Selection has changed while at least one button is down. Take down current CURSEL highlighting, switch to
                ;; NEWSEL.  If the mouse condition is removed, the secondary selection can be lost if the mouse moves while the
                ;; operation keys are still down.  But if the copy isn't done when NEWSEL picks out an object, the object will be
                ;; lost.

                (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                (\TEDIT.COPYSEL NEWSEL CURSEL)
                (\TEDIT.SHOWSEL CURSEL T TEXTOBJ)))

    ;; CURSEL now matches the display and CUROPERATION.

finally ;; Out of Polling loop
        (CL:UNLESS (FGETSEL NEWSEL SET)
```

```
                        ;; .Here to restore when no valid selection, maybe an unhappy image object?
                        (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)      ; Turn off CURSEL
                        (\TEDIT.SET.SEL.LOOKS (TEXTSEL TEXTOBJ)
                               'NORMAL)                          ; Restore TEXTSEL
                        (\TEDIT.SHOWSEL NIL T TEXTOBJ)
                        (RETURN))
                    (\TEDIT.BUTTONEVENTFN.DOOPERATION CURSEL CUROPERATION TSTREAM PANE PENDINGDEL TTYPROC))))]
)
```

(\TEDIT.BUTTONEVENTFN.DOOPERATION

```
  [LAMBDA (CURSEL CUROPERATION TSTREAM PANE PENDINGDEL TTYPROC)  ; Edited 25-Nov-2024 22:22 by rmk
                                                                ; Edited  4-Nov-2024 13:09 by rmk
                                                                ; Edited  3-Nov-2024 07:20 by rmk
                                                                ; Edited 17-Oct-2024 22:01 by rmk
                                                                ; Edited  7-Oct-2024 08:31 by rmk
                                                                ; Edited  4-Oct-2024 10:26 by rmk
                                                                ; Edited  2-Oct-2024 10:01 by rmk
                                                                ; Edited 12-Sep-2024 17:46 by rmk
                                                                ; Edited 10-Sep-2024 17:39 by rmk
                                                                ; Edited 27-Aug-2024 14:35 by rmk
                                                                ; Edited 24-Aug-2024 00:11 by rmk
                                                                ; Edited 19-Jul-2024 23:50 by rmk
                                                                ; Edited 22-Apr-2024 23:52 by rmk
                                                                ; Edited 19-Feb-2024 00:13 by rmk
    ;; Executes CUROPERATION in the given TTY process.  Calls FOREIGN.COPY if the TTY process is not a Tedit process (either this one or
    ;; another one).

    ;; NOTE:  TTYPROC is passed in so that you can break or HELP without it changing to the break window.

    ;; On entry, CURSEL's highlighting is on the screen

    (CL:WHEN (FGETSEL CURSEL SET)
        (LET* ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
               (TTYW (PROCESSPROP TTYPROC 'WINDOW))
               (TTYSTREAM (AND TTYW (fetch (TEXTWINDOW WTEXTSTREAM) of TTYW)))
               [TTYSEL (AND TTYSTREAM (TEXTSEL (GETTSTR TTYSTREAM TEXTOBJ]
               (SELFN (GETTEXTPROP TEXTOBJ 'SELFN))
               (TEXTSEL (TEXTSEL TEXTOBJ)))
            ;; TTYSTREAM guaranteed EQ to TSTREAM for NORMAL and DELETE. TTYSEL is NIL for foreign copy.
            (SELECTQ CUROPERATION
                (NORMAL (\TEDIT.COPYSEL CURSEL TEXTSEL)
                        (FSETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ TEXTSEL))
                        (if (FGETSEL TEXTSEL SELOBJ)
                            then (\TEDIT.OPERATE.OBJECT TSTREAM TEXTSEL PANE 'SELECTED)
                          else (FSETTOBJ TEXTOBJ BLUEPENDINGDELETE PENDINGDEL)
                               (\TEDIT.SETCARET TEXTSEL PANE TEXTOBJ T)))
                (DELETE ;; \TEDIT.DELETE converts TTYSEL (= TEXTSEL) to a point-caret.
                        (\TEDIT.COPYSEL CURSEL TEXTSEL)
                        (CL:WHEN (\TEDIT.DELETE TEXTOBJ TEXTSEL)
                                                                ; Make sure the caret blinks in the position of a successful
                                                                ; deletion
                            (FSETSEL TEXTSEL HASCARET T))
                        (\TEDIT.SETCARET TEXTSEL PANE TEXTOBJ T))
                (COPY (CL:IF TTYSEL
                          (\TEDIT.COPY CURSEL TTYSEL TSTREAM TTYSTREAM)
                          (\TEDIT.FOREIGN.COPY TTYW CURSEL TSTREAM))
                      (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ))
                (MOVE (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                      (if TTYSEL
                          then (\TEDIT.MOVE CURSEL TTYSEL TSTREAM TTYSTREAM)
                        else (\TEDIT.FOREIGN.COPY TTYW CURSEL TSTREAM)
                                                                ; TEXTSEL moves to deletion point
                             (\TEDIT.UPDATE.SEL TEXTSEL (FGETSEL CURSEL CH#)
                                    0
                                    'RIGHT)
                             (\TEDIT.DELETE TEXTOBJ CURSEL)
                             (\TEDIT.SHOWSEL TEXTSEL T TEXTOBJ)))
                (COPYLOOKS (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                           (if TTYSEL
                               then (if (EQ 'PARA (FGETSEL CURSEL SELKIND))
                                        then (\TEDIT.CHANGE.PARALOOKS TTYSTREAM (PPARALOOKS
                                                                                  (\TEDIT.CHTOPC (FGETSEL
                                                                                                   CURSEL
                                                                                                   CH#)
                                                                                                 TEXTOBJ))
                                                                      TTYSEL)
                                      else (\TEDIT.CHANGE.CHARLOOKS TTYSTREAM (PCHARLOOKS (\TEDIT.CHTOPC
                                                                                           (FGETSEL CURSEL CH#)
                                                                                           TEXTOBJ))
                                                                    TTYSEL))
                             else (if (EQ 'PARA (FGETSEL CURSEL SELKIND))
                                      then (\TEDIT.CHANGE.PARALOOKS TSTREAM (PPARALOOKS (\TEDIT.CHTOPC
                                                                                         (FGETSEL CURSEL CH#)
                                                                                         TEXTOBJ))
                                                                    CURSEL)
```

```
                                    else (\TEDIT.CHANGE.CHARLOOKS TSTREAM (PCHARLOOKS (\TEDIT.CHTOPC
                                                                                       (FGETSEL CURSEL CH#)
                                                                                       TEXTOBJ))
                                                                  CURSEL))))
                (\TEDIT.THELP "Bad selection operation" CUROPERATION))
            (CL:UNLESS PENDINGDEL
                (\TEDIT.SET.SEL.LOOKS (TEXTSEL TEXTOBJ)
                        'NORMAL))
            (CL:WHEN SELFN                                  ; Maybe for logging of selections?
                (APPLY* SELFN TEXTOBJ CURSEL CUROPERATION 'FINAL))
            (CL:UNLESS (FGETTOBJ TEXTOBJ MENUFLG)           ; Globals are documented, unfortunately.
                (\TEDIT.SET.GLOBAL.SELECTIONS CUROPERATION CURSEL))))])
```

(\TEDIT.BUTTONEVENTFN.GETOPERATION

```
[LAMBDA (READONLY CUROPERATION)                             ; Edited 30-Sep-2024 08:34 by rmk
                                                            ; Edited 25-Sep-2024 11:48 by rmk
                                                            ; Edited 23-Sep-2024 22:29 by rmk
                                                            ; Edited 11-Sep-2024 14:45 by rmk
                                                            ; Edited 27-Aug-2024 22:29 by rmk
                                                            ; Edited 19-Aug-2024 00:47 by rmk
                                                            ; Edited 20-Jul-2024 13:01 by rmk
                                                            ; Edited 27-Jan-2024 12:55 by rmk
```

;; Look at the mode keys to figure out the new operation.

;; In a read-only document you cannot move or delete. Selection is NORMAL if no keys are down.

```
(CL:WHEN [AND (EQ 'MOVE CUROPERATION)
              (NEQ (SHIFTDOWNP 'SHIFT)
                   (SHIFTDOWNP 'CTRL]
```

;; If both were down (MOVE) and now they're in different states, wait a bit to see what the other one will do.

```
    (DISMISS 50 NIL T))
(if (AND (SHIFTDOWNP 'CTRL)
         (NOT READONLY))
    then (if (SHIFTDOWNP 'SHIFT)
             then 'MOVE
           else 'DELETE)
  elseif (OR (SHIFTDOWNP 'SHIFT)
             (KEYDOWNP 'COPY))
    then 'COPY
  elseif (SHIFTDOWNP 'META)
    then 'COPYLOOKS
  else                                                      ; No keys down
       'NORMAL])
```

(\TEDIT.BUTTONEVENTFN.CURSEL.INIT

```
[LAMBDA (NEWOPERATION TEXTOBJ)                              ; Edited 30-Nov-2024 15:45 by rmk
                                                            ; Edited 27-Nov-2024 20:23 by rmk
                                                            ; Edited 22-Oct-2024 23:10 by rmk
                                                            ; Edited 20-Oct-2024 23:38 by rmk
                                                            ; Edited  2-Oct-2024 09:59 by rmk
                                                            ; Edited 12-Sep-2024 17:14 by rmk
```

;; Create and initialize CURSEL according to NEWOPERATION. Start with copy of TEXTSEL so line-chains correspond to the number of split
;; panes. TEXTSEL maybe taken down but is otherwise not modified.

;; NILvalue signals abort

```
(PROG [(CURSEL (\TEDIT.COPYSEL (TEXTSEL TEXTOBJ]
      (SELECTQ NEWOPERATION
           (NORMAL ;; Operating in this document. Our initial CURSEL is consistent with TEXTSEL and display.
                   (FSETSEL (TEXTSEL TEXTOBJ)
                          ONFLG NIL)                        ; Transferred display status to CURSEL, restore later if needed
                   (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)      ; Take down current hilight
                   (if (\TEDIT.MOUSESTATE RIGHT)
                       then ;; Extending the current selection: coerce to PENDINGDEL/black
                            (\TEDIT.SET.SEL.LOOKS CURSEL 'PENDINGDEL)
                     elseif (FGETTOBJ TEXTOBJ BLUEPENDINGDELETE)
                       then ;; Not extending: turn off BPD highlighting and reduce to a point selection at the caret.
                            (FSETTOBJ TEXTOBJ BLUEPENDINGDELETE NIL)
                            (\TEDIT.UPDATE.SEL CURSEL (TEDIT.GETPOINT TEXTOBJ CURSEL)
                                   0 NIL 'NORMAL)
                            (\TEDIT.FIXSEL CURSEL TEXTOBJ)
                            (\TEDIT.SHOWSEL CURSEL T TEXTOBJ))
                   (\TEDIT.SHOWSEL CURSEL T TEXTOBJ))
           (DELETE ;; Deleting (CTRL) somewhere else. Turn off CURSEL's highlighting, which was transferred from TEXTSEL
                   (\TEDIT.SHOWSEL CURSEL NIL TEXTOBJ)
                   (\TEDIT.SET.SEL.LOOKS CURSEL 'DELETE))
           ((MOVE COPY COPYLOOKS) ;; Source text from here, TTY target maythat be here, another Tedit, or foreign. TEXTSEL remains
                                  ;; visible
               (CL:WHEN (\TEDIT.MOUSESTATE RIGHT)           ; Funny to copy while extending
                   (RETURN))
```

```
                (\TEDIT.SET.SEL.LOOKS CURSEL NEWOPERATION)
                (FSETSEL CURSEL SET NIL))
            (FSETSEL CURSEL SET NIL))
        (RETURN CURSEL])
```

(\TEDIT.BUTTONEVENTFN.INACTIVE

```
[LAMBDA (TEXTOBJ PANE)                                  ; Edited 24-Apr-2024 09:45 by rmk
                                                        ; Edited 16-Mar-2024 00:22 by rmk
                                                        ; Edited  9-Feb-2024 00:00 by rmk
                                                        ; Edited 27-Jan-2024 11:40 by rmk
```

;; TEXTOBJ is the textobj associated with some window and presumably therefore has (or had) an associated editing process. This returns T if the ;; session is currently inactive or if this TEXTOBJ or TEXTOBJ cannot still be used as a source of information. If inactive, this also either executes ;; the generic window operations (if RIGHT is down, whether or not in the title region) or perhaps reestablishes the process.

;; If EDITOPACTIVE, something else is going on that we don't want to interfere with

```
(if [AND (NOT (FGETTOBJ TEXTOBJ EDITFINISHEDFLG))
         (NOT (FGETTOBJ TEXTOBJ EDITOPACTIVE))
         (OR (WINDOWPROP PANE 'PROCESS)
             (GETTOBJ TEXTOBJ TXTREADONLY)
             (SHIFTDOWNP 'SHIFT)
             (SHIFTDOWNP 'CTRL)
             (SHIFTDOWNP 'META)
             (KEYDOWNP 'MOVE)
             (KEYDOWNP 'COPY]
    then NIL
  elseif (\TEDIT.MOUSESTATE RIGHT)
    then ;; Right button anywhere in a dead window gets the window command menu. Window is still inactive
         (DOWINDOWCOM PANE)
         T
  elseif [AND (NOT (GETTEXTPROP TEXTOBJ 'READONLY))
              (NOT (GETTEXTPROP TEXTOBJ 'SELECTONLY))
              [NOT (PROCESSP (WINDOWPROP PANE 'PROCESS]
              (OR T (AND (\TEDIT.MOUSESTATE MIDDLE)
                         (EQ 'NewEditProcess (MENU (CREATE MENU
                                                           ITEMS _ '(NewEditProcess]
    then ;; Why do we need a Middle-button menu to restart a dead window.  If it's clicked in, just restart it.
         (SETTOBJ TEXTOBJ EDITOPACTIVE NIL)
         (TEDIT (fetch (TEXTWINDOW WTEXTSTREAM) of PANE)
                PANE)
         NIL])
```

(\TEDIT.BUTTONEVENTFN.INTITLE

```
[LAMBDA (Y PANE TEXTOBJ)                                ; Edited  1-Dec-2024 12:02 by rmk
                                                        ; Edited 13-Jun-2024 22:10 by rmk
                                                        ; Edited 27-Jan-2024 10:42 by rmk
```

;; Special behavior if Y is the title region of PANE?

```
(LET (USERFN)
     (CL:WHEN (IGREATERP Y (PANETOP PANE))
         [if (\TEDIT.MOUSESTATE RIGHT)
             then (DOWINDOWCOM PANE)
           elseif (AND (OR (SHIFTDOWNP 'SHIFT)
                           (KEYDOWNP 'COPY))
                       (MOUSESTATE LEFT))
             then (bind THING unless (OR (SHIFTDOWNP 'SHIFT)
                                         (KEYDOWNP 'COPY))
                     do (GETMOUSESTATE)
                        (CL:UNLESS (INSIDEP (PANEREGION PANE)
                                            (LASTMOUSEX PANE)
                                            (LASTMOUSEY PANE))
                            (CL:WHEN [SETQ THING (OR (GETTOBJ TEXTOBJ TXTFILE)
                                                     (GETTEXTPROP TEXTOBJ 'ITEM-NAME]
                                (COPYINSERT (CL:IF (STREAMP THING)
                                                   (MKSTRING (FULLNAME THING))
                                                   THING))))
                        (RETURN))
           elseif (MOUSESTATE (OR LEFT MIDDLE))
             then (CL:WHEN (AND (SETQ USERFN (WINDOWPROP PANE 'TEDIT.TITLEMENUFN))
                                (NEQ USERFN 'DON'T))
                           (ADD.PROCESS (LIST USERFN (KWOTE PANE))))]
         T)])
```

(\TEDIT.COPYINSERTFN

```
[LAMBDA (INSERTION PANE)                                ; Edited 27-Aug-2024 10:38 by rmk
                                                        ; Edited  7-Jul-2024 09:26 by rmk
                                                        ; Edited 29-Apr-2024 13:37 by rmk
                                                        ; Edited 22-Apr-2024 23:47 by rmk
                                                        ; Edited 17-Feb-2024 12:52 by rmk
```

;; The COPYINSERTFN of Tedit windows. INSERTION is inserted into the TSTREAM of PANE.

;; IRM says that it should use BKSYSBUF for strings.

```
    (LET ((TSTREAM (TEXTSTREAM PANE)))
         (for I inside INSERTION do (if (IMAGEOBJP I)
                                        then (TEDIT.INSERT.OBJECT I TSTREAM)
                                      elseif (OR (STRINGP I)
                                                 (LITATOM I))
                                        then (TEDIT.INSERT TSTREAM I])
```

(\TEDIT.FOREIGN.COPY

```
  [LAMBDA (TTYW SOURCESEL SOURCESTREAM BKSYSBUFP)                    ; Edited 27-Aug-2024 13:38 by rmk
                                                                     ; Edited  7-Jul-2024 09:26 by rmk
                                                                     ; Edited 29-Apr-2024 13:37 by rmk
                                                                     ; Edited 22-Apr-2024 23:47 by rmk
                                                                     ; Edited 17-Feb-2024 12:52 by rmk
    ;; Inserts the information in SOURCESEL into a non-Tedit TTY stream.

    (CL:WHEN (IGREATERP (GETSEL SOURCESEL DCH)
                        0)                                           ; If empty, nothing to do
        [if (AND NIL (NOT BKSYSBUFP)
                 (WINDOWPROP TTYW 'COPYINSERTFN))
            then ;; This is a stub for a definition that knows how to do a looked string object, given that the destination TTY window has a
                 ;; COPYINSERTFN.  OBJECTFROMSEL is in {LFG}tedit/UNBREAKABLESTRING

                 (COPYINSERT (OBJECTFROMSEL SOURCESEL))
          else ;; Have to go character by character because COPYINSERT does (PRIN2 BKSYSBUF), which creates undesired string quotes.
               (for CHNO CH from (FGETSEL SOURCESEL CH#) to (SUB1 (FGETSEL SOURCESEL CHLIM))
                  while (SETQ CH (TEDIT.NTHCHARCODE SOURCESTREAM CHNO)) do ;; Maybe should apply the preprintfn ?
                                                                           (CL:IF (IMAGEOBJP CH)
                                                                               (COPYINSERT CH)
                                                                               (BKSYSBUF (CHARACTER CH)))])])

)

(DEFINEQ
```

(\TEDIT.PANE.SPLIT

```
  [LAMBDA (TEXTOBJ WINDOWTOSPLIT)                                    ; Edited 23-Oct-2024 09:50 by rmk
                                                                     ; Edited 21-Oct-2024 00:33 by rmk
                                                                     ; Edited 27-Jan-2024 11:39 by rmk
                                                                     ; Edited  1-Oct-2023 23:30 by rmk
                                                                     ; Edited 12-Oct-2021 15:01 by rmk:
    ;; If in the split region, determine and execute the splitting operations for PANE.

    (CL:WHEN (EQ (GETTOBJ TEXTOBJ MOUSEREGION)
                 'PANE)                                              ; In the split/ops region
        [LET ([WINDOWOPREGION (create REGION
                                      LEFT _ (DIFFERENCE (fetch (TEXTOBJ WRIGHT) of TEXTOBJ)
                                                         \TEDIT.OP.WIDTH)
                                      BOTTOM _ \TEDIT.OP.BOTTOM
                                      WIDTH _ \TEDIT.OP.WIDTH
                                      HEIGHT _ (fetch (REGION HEIGHT) of (WINDOWPROP WINDOWTOSPLIT 'REGION]
              Y OPERATION)
             [while [AND (MOUSESTATE (OR LEFT MIDDLE RIGHT))
                         (INSIDE? WINDOWOPREGION (LASTMOUSEX WINDOWTOSPLIT)
                                  (SETQ Y (LASTMOUSEY WINDOWTOSPLIT]
                do ;; Wait until he lets up on a button, and signal which button was last pushed.
                   (BLOCK)
                   (COND
                      ((MOUSESTATE MIDDLE)
                       (CURSOR \TEDIT.MAKESPLITCURSOR)
                       (SETQ OPERATION 'SPLIT))
                      ((MOUSESTATE LEFT)
                       (CURSOR \TEDIT.MOVESPLITCURSOR)
                       (SETQ OPERATION 'MOVE))
                      ((MOUSESTATE RIGHT)
                       (CURSOR \TEDIT.UNSPLITCURSOR)
                       (SETQ OPERATION 'UNSPLIT]
             (COND
                ((INSIDE? WINDOWOPREGION (LASTMOUSEX WINDOWTOSPLIT)
                          (SETQ Y (LASTMOUSEY WINDOWTOSPLIT)))
                 (CURSOR \TEDIT.SPLITCURSOR)
                 (SELECTQ OPERATION
                      (SPLIT                                         ; Splitting the window
                             (\TEDIT.SPLITW WINDOWTOSPLIT Y))
                      (UNSPLIT                                       ; Rejoining two panes
                               (\TEDIT.UNSPLITW WINDOWTOSPLIT))
                      (MOVE                                          ; Moving the divider between two panes.
                            (TEDIT.PROMPTPRINT TEXTOBJ "Split-point moving is not yet implemented" T T))
                      (\TEDIT.THELP)))
                (T (CURSOR T]
        T)])
```

(\TEDIT.SPLITW

```
[LAMBDA (OLDPANE Y)                                                    ; Edited  1-Dec-2024 11:27 by rmk
                                                                       ; Edited 20-Nov-2024 12:37 by rmk
                                                                       ; Edited 17-Nov-2024 18:59 by rmk
                                                                       ; Edited  5-Jul-2024 11:37 by rmk
                                                                       ; Edited 30-Jun-2024 21:59 by rmk
                                                                       ; Edited 28-Jun-2024 21:08 by rmk
                                                                       ; Edited 21-Jun-2024 22:47 by rmk
                                                                       ; Edited 19-Jun-2024 08:57 by rmk
                                                                       ; Edited 17-Jun-2024 09:01 by rmk
                                                                       ; Edited 13-Jun-2024 17:34 by rmk
                                                                       ; Edited 18-May-2024 16:24 by rmk
                                                                       ; Edited 24-Apr-2024 09:42 by rmk
                                                                       ; Edited  5-May-2024 23:13 by rmk
                                                                       ; Edited 20-Mar-2024 11:01 by rmk
                                                                       ; Edited  8-Feb-2024 23:38 by rmk
                                                                       ; Edited  2-Jan-2024 19:21 by rmk
                                                                       ; Edited  4-Oct-2023 10:37 by rmk
                                                                       ; Edited  5-Nov-2022 23:51 by rmk
                                                                       ; Edited 30-May-91 23:38 by jds
    ;; Split window OLDPANE at window-relelative Y into 2 panes that can scroll independently.

    ;; Note that TSTREAM and TEXTOBJ are the same for all panes.

    ;; Original code was goofy: after carefully setting things up, attached menus and prompts would move into the main-window space.  Setting and
    ;; reseting the ATTACHEDWINDOWS property seems to fix that.

    (LET* ((WREG (WINDOWPROP OLDPANE 'REGION))
           (TSTREAM (fetch (TEXTWINDOW WTEXTSTREAM) of OLDPANE))
           (TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
           (SEL (TEXTSEL TEXTOBJ))
           (NEXTPANE (GETPANEPROP (PANEPROPS OLDPANE)
                                  NEXTPANE))
           ATTACHEDWINDOWS NEWPANE PROPS NEXTCHAR1)
          (CL:UNLESS Y                                                 ; Y-position of the split, either supplied or mouse.
              (SETQ Y (LASTMOUSEY OLDPANE)))
          (CL:WHEN NEXTPANE                                            ; If there's already a pane below this one, detach it for the
                                                                       ; moment.
              (DETACHWINDOW NEXTPANE))
          (SETQ ATTACHEDWINDOWS (WINDOWPROP OLDPANE 'ATTACHEDWINDOWS NIL))

          ;; Reshape the original window to form the upper pane. This fixes/displays the current selection in all existing panes

          (SHAPEW OLDPANE (create REGION using WREG BOTTOM _ (IPLUS (fetch BOTTOM of WREG)
                                                                    Y)
                                                    HEIGHT _ (IDIFFERENCE (fetch HEIGHT of WREG)
                                                                          Y)))

          ;; OLDPANE has now been shrunk, redisplayed with new lines, and highlighted.  The selection is on.

          ;; Attach the new window, without disturbing the pre-existing attached windows

          (SETQ NEWPANE (CREATEW (create REGION using WREG HEIGHT _ Y)))
          (ATTACHWINDOW NEWPANE OLDPANE 'BOTTOM 'JUSTIFY 'MAIN)        ; and attach a lower pane.
          [WINDOWPROP OLDPANE 'ATTACHEDWINDOWS (APPEND ATTACHEDWINDOWS (WINDOWPROP OLDPANE 'ATTACHEDWINDOWS]

          ;; [end of attached-window hackery to prevent disturbance while short]

          ;;

          (WINDOWPROP NEWPANE 'TEDITCREATED T)
          (DSPFONT (GETCLOOKS (FGETTOBJ TEXTOBJ CARETLOOKS)
                              CLFONT)
                 NEWPANE)                                              ; Set the font on the display stream to be the current one from
                                                                       ; CARETLOOKS

          ;; Not sure if same PROPS as for OLDPANE (which this would inherit from primary window)

          [SETQ PROPS (APPEND '(NOTITLE T PROMPTWINDOW DON'T TITLEMENUFN NILL)
                              (COPY (FGETTOBJ TEXTOBJ EDITPROPS]
          (\TEDIT.MINIMAL.WINDOW.SETUP NEWPANE TSTREAM PROPS OLDPANE)

          ;; Insert L1 and LN cells for NEWPANEafter OLDPANE's cells in each selection.  The selections were created when the original textsteam
          ;; was opened.

          ;; Create the first line of NEWPANE starting at the character just after the last line of the now-shrunken OLDPANE.

          [SETQ NEXTCHAR1 (for L (BOTTOM _ (PANEBOTTOM OLDPANE))
                             inlines
                             (PANEPREFIX OLDPANE) unless (AND (FGETLD L NEXTLINE)
                                                              (IGEQ (FGETLD (FGETLD L NEXTLINE)
                                                                            YBOT)
                                                                    BOTTOM))
                           do ;; If we run off the end of the text, start with at least the last line (which may just be EOL's).
                              (RETURN (if (AND (IGEQ (FGETLD L LCHAR1)
                                                     (TEXTLEN TEXTOBJ))
                                               (FGETLD L PREVLINE))
                                          then (FGETLD (FGETLD L PREVLINE)
                                                       LCHAR1)
                                          else (FGETLD L LCHARLIM]
          (\TEDIT.WINDOW.SETUP NEWPANE TSTREAM PROPS OLDPANE NEXTCHAR1)
                                                                       ; OLDPANE covers everything before
          (WINDOWPROP NEWPANE 'PROCESS (WINDOWPROP OLDPANE 'PROCESS))
          (CL:WHEN (GETSEL SEL ONFLG)
              (SETSEL SEL ONFLG NIL)                                   ; Turn it off, so we can turn it on for NEWPANE
```

```
            (\TEDIT.SHOWSEL SEL T TEXTOBJ NEWPANE))          ; Tell NEWPANE about the old pane below it
          (CL:WHEN NEXTPANE                                  ; There was already a pane below this one. Attach it to the new
                                                             ; lower pane.
            (ATTACHWINDOW NEXTPANE NEWPANE 'BOTTOM 'JUSTIFY 'MAIN))])
```

(\TEDIT.UNSPLITW

```
[LAMBDA (PANE)                                               ; Edited  1-Jul-2024 08:50 by rmk
                                                             ; Edited 29-Jun-2024 09:00 by rmk
                                                             ; Edited 18-May-2024 16:21 by rmk
                                                             ; Edited 12-May-2024 20:58 by rmk
                                                             ; Edited 15-Mar-2024 18:30 by rmk
                                                             ; Edited 20-Mar-2024 11:01 by rmk
                                                             ; Edited 21-Feb-2024 08:31 by rmk
                                                             ; Edited 11-Feb-2024 11:14 by rmk
                                                             ; Edited  2-Jan-2024 21:11 by rmk
                                                             ; Edited 30-Sep-2023 14:17 by rmk
                                                             ; Edited 21-Sep-2023 09:02 by rmk
                                                             ; Edited  2-Sep-2023 16:18 by rmk
                                                             ; Edited 18-Apr-2023 23:41 by rmk
                                                             ; Edited  6-Nov-2022 00:06 by rmk
  (PROG* ((TSTREAM (fetch (TEXTWINDOW WTEXTSTREAM) of PANE))
          (TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))
          (SEL (FGETTOBJ TEXTOBJ SEL))
          PREVPANE NEXTPANE ATTACHEDWINDOWS)
         (CL:WHEN (EQ PANE (FGETTOBJ TEXTOBJ PRIMARYPANE))
                (RETURN))
         (SETQ PREVPANE (GETPANEPROP (PANEPROPS PANE)
                                     PREVPANE))
         (SETQ NEXTPANE (GETPANEPROP (PANEPROPS PANE)
                                     NEXTPANE))
         (FSETTOBJ TEXTOBJ SELPANE (FGETTOBJ TEXTOBJ PRIMARYPANE))
         (for P inpanes TEXTOBJ as SL1 in (GETSEL SEL L1) as SLN in (GETSEL SEL LN)
            when (EQ PANE P) do (change (GETSEL SEL L1)
                                        (DREMOVE SL1 DATUM))
                                (change (GETSEL SEL LN)
                                        (DREMOVE SLN DATUM))
                                (RETURN))
         (WINDOWPROP PANE 'CURSOROUTFN NIL)
         (WINDOWPROP PANE 'CURSORMOVEDFN NIL)
         (\TEDIT.UNLINKPANE PANE)                            ; Disconnect
    ;;
    ;; Done with the deleted pane, assign its region to the pane above and redisplay.
    ;; Now rearrange the pane window-attachment linkages. This gives PANE's region to its prior pane.
    ;; Original code moved the promptwindow and attached menus down into the region of the main window, shrinking the overall footprint.  This code
    ;; only unsplits the target pane, leaving everything else unchanged.
         (DETACHWINDOW PANE)
         (SETQ ATTACHEDWINDOWS (WINDOWPROP PREVPANE 'ATTACHEDWINDOWS NIL))
         [SHAPEW PREVPANE (UNIONREGIONS (WINDOWPROP PANE 'REGION)
                                        (WINDOWPROP PREVPANE 'REGION]
         (WINDOWPROP PREVPANE 'ATTACHEDWINDOWS ATTACHEDWINDOWS)
         (CL:WHEN NEXTPANE
             ;; PANE had a yet lower pane attached to it.  Promote it to PANE's position in the attachment chain
             (DETACHWINDOW NEXTPANE)
             (ATTACHWINDOW NEXTPANE PREVPANE 'BOTTOM 'JUSTIFY 'MAIN))
         (CLOSEW PANE])
```

(\TEDIT.LINKPANES

```
[LAMBDA (PANE1 PANE2)                                        ; Edited  1-Jul-2024 08:39 by rmk
                                                             ; Edited 29-Jun-2024 00:12 by rmk
  ;; Splices PANE2 into the pane sequence after existing PANE1
  (LET ((PPROPS1 (PANEPROPS PANE1))
        (PPROPS2 (PANEPROPS PANE2))
        PANE1NEXT)
       (SETQ PANE1NEXT (GETPANEPROP PPROPS1 NEXTPANE))
       (SETPANEPROP PPROPS2 PREVPANE PANE1)
       (SETPANEPROP PPROPS2 NEXTPANE (GETPANEPROP PPROPS1 NEXTPANE))
       (SETPANEPROP PPROPS1 NEXTPANE PANE2)
       (CL:WHEN PANE1NEXT
           (SETPANEPROP (PANEPROPS PANE1NEXT)
                  PREVPANE PANE2))
      PANE2])
```

(\TEDIT.UNLINKPANE

```
[LAMBDA (PANE)                                               ; Edited 28-Jun-2024 23:47 by rmk
  ;; Removes PANE from its PANE sequence
  (LET ((PANEPROPS (PANEPROPS PANE))
        NEXT PREV)
       (SETQ NEXT (GETPANEPROP PANEPROPS NEXTPANE))
       (SETQ PREV (GETPANEPROP PANEPROPS PREVPANE))
```

```
         (CL:WHEN PREV
             (SETPANEPROP (PANEPROPS PREV)
                    NEXTPANE NEXT))
         (CL:WHEN NEXT
             (SETPANEPROP (PANEPROPS NEXT)
                    PREVPANE PREV))
         PREV])

)

(MOVD? 'NILL 'GRAB-TYPED-REGION)

(MOVD? 'NILL 'REGISTER-TYPED-REGION)

(RPAQ? \TEDIT.OP.WIDTH 12)

(RPAQ? \TEDIT.OP.BOTTOM 12)

(RPAQ? \TEDIT.LINEREGION.WIDTH 12)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \TEDIT.OP.WIDTH \TEDIT.OP.BOTTOM \TEDIT.LINEREGION.WIDTH)
)
)

(RPAQ BXCARET (CURSORCREATE '
                          'NIL 3 4))

(RPAQ BXHICARET (CURSORCREATE '
                        'NIL 4 7))

(RPAQ \TEDIT.LINECURSOR (CURSORCREATE '
                                 'NIL 15 15))

(RPAQ \TEDIT.SPLITCURSOR (CURSORCREATE '
                                  'NIL 4 4))

(RPAQ \TEDIT.MOVESPLITCURSOR (CURSORCREATE '
                                      'NIL 4 4))

(RPAQ \TEDIT.UNSPLITCURSOR (CURSORCREATE '
                                    'NIL 4 4))

(RPAQ \TEDIT.MAKESPLITCURSOR (CURSORCREATE '
                                      'NIL 4 4))
```

;; User-level "is this a TEdit window?" function.

```
(DEFINEQ
```

(TEDITWINDOWP

```
  [LAMBDA (WINDOW)                                          ; Edited 28-Jun-2024 22:16 by rmk
                                                            ; Edited 25-Jun-2024 11:59 by rmk
                                                            ; Edited 22-Jan-2024 10:57 by rmk
                                                            ; Edited 15-Sep-2023 21:03 by rmk
                                                            ; Edited 16-Jan-89 10:28 by jds
```

;; If WINDOW is or denotes the window of a text stream, returns that textstream's primary pane. The test is that the returned window has a ;; TEXTOBJ property, and the TEXTOBJ thinks this is its primary pane.

```
    (LET ((TEXTOBJ (TEXTOBJ WINDOW T)))
         (CL:WHEN TEXTOBJ (GETTOBJ TEXTOBJ PRIMARYPANE])

)
```

;; User-typein support

```
(DEFINEQ
```

(TEDIT.GETINPUT

```
  [LAMBDA (STREAM PROMPTSTRING DEFAULTSTRING DELIMITER.LIST)      ; Edited 22-Jan-2024 00:09 by rmk
                                                                  ; Edited 22-Sep-2023 19:57 by rmk
                                                                  ; Edited 30-Jul-2023 08:51 by rmk
                                                                  ; Edited 21-Jan-2022 23:14 by rmk
                                                                  ; Edited 30-May-91 23:34 by jds
```

;; Ask for input (file names, &c) for TEdit, perhaps with a default.

```
(LET* ((TEXTOBJ (TEXTOBJ STREAM))
       (TPROMPT (GETTOBJ TEXTOBJ PROMPTWINDOW))
       RESULT)
      (SETQ TPROMPT (SELECTQ TPROMPT
                        (DON'T [COND
                                   ((GETTEXTPROP TEXTOBJ 'PWINDOW.ON.DEMAND)
                                    (GETPROMPTWINDOW (\TEDIT.MAINW STREAM])
                        (NIL [GETPROMPTWINDOW (\TEDIT.MAINW STREAM)
                                    NIL NIL (NOT (GETTEXTPROP TEXTOBJ 'PWINDOW.ON.DEMAND])
                        TPROMPT))
      (COND
         (TPROMPT                                          ; If it's our own promptwindow, just clear it.
                (CLEARW TPROMPT)
                (FRESHLINE TPROMPT))
         (T                                                ; If it's the system's window, just move to a new line.
            (FRESHLINE PROMPTWINDOW)))
      (SETQ RESULT (TTYINPROMPTFORWORD PROMPTSTRING DEFAULTSTRING NIL (OR TPROMPT PROMPTWINDOW)
                        NIL
                        'TTY
                        (OR DELIMITER.LIST (CHARCODE (EOL LF TAB ESCAPE ^X)))
                        NIL))
      (CL:WHEN (AND (EQ (CHARCODE %")
                        (CHCON1 RESULT))
                    (EQ (CHARCODE %")
                        (NTHCHARCODE RESULT -1)))
          ;; Presumably it is not intended to have a string with string quotes on the edges.
          (SETQ RESULT (SUBSTRING RESULT 2 -2))
          (WINDOWPROP (OR TPROMPT PROMPTWINDOW)
                 'PROCESS NIL))
      RESULT])
```

(\TEDIT.MAKEFILENAME

```
[LAMBDA (STRING)
```

;; Edited 29-Jun-2024 16:26 by rmk

;; Edited 18-Dec-2023 22:45 by rmk

;; Edited 9-Sep-2023 17:13 by rmk

;; Edited 24-Oct-2022 00:02 by rmk: Originally returned an atom, which is no longer a valid filename

(* jds "8-Feb-85 11:25")

;; Removes leading and trailing spaces from a non-NIL string

```
(CL:UNLESS (STRING.EQUAL STRING NIL)
    (CL:STRING-TRIM '(#\Space)
          STRING)
    (CL:UNLESS (EQ 0 (NCHARS STRING))
          STRING))])

)

;; Attached Prompt window support.

(DEFINEQ
```

(TEDIT.PROMPTWINDOW

```
[LAMBDA (TSTREAM)                                          ; Edited 18-Mar-2024 17:48 by rmk
```

;; Return the TEdit promptwindow, if any, from a TEdit thing (textstream, textobj, or TEdit window).

```
(FGETTOBJ (TEXTOBJ TSTREAM)
       PROMPTWINDOW])
```

(TEDIT.PROMPTPRINT

```
[LAMBDA (TEXTSTREAM MSG CLEAR? FLASH?)                     ; Edited 29-Dec-2024 14:45 by rmk
                                                           ; Edited 26-Nov-2023 10:10 by rmk
                                                           ; Edited 10-Sep-2023 00:27 by rmk
                                                           ; Edited 30-Jul-2023 08:52 by rmk
                                                           ; Edited 9-Jul-2023 12:37 by rmk
                                                           ; Edited 5-Apr-2023 15:08 by rmk
                                                           ; Edited 4-Jun-93 12:04 by sybalsky:mv:envos
```

;; Print a message in the editor's prompt window (if none, use the global promptwindow). Optionally clear the window first.

```
(LET ((TEXTOBJ (TEXTOBJ TEXTSTREAM T))
      PWINDOW MAINWINDOW)
     (if TEXTOBJ
         then (CL:WHEN (SETQ MAINWINDOW (\TEDIT.MAINW TEXTOBJ))
                  [SETQ PWINDOW (CAR (NLSETQ (SELECTQ PWINDOW
                                                 (DON'T (CL:WHEN (GETTEXTPROP TEXTOBJ 'PWINDOW.ON.DEMAND)
                                                                 (GETPROMPTWINDOW MAINWINDOW)))
                                                 (NIL (CL:WHEN TEXTSTREAM
                                                          [GETPROMPTWINDOW MAINWINDOW NIL NIL
                                                                 (NOT (GETTEXTPROP TEXTOBJ
                                                                          'PWINDOW.ON.DEMAND]))
```

```
                                                    PWINDOW])   ; Try to find an editor's prompt window for our message
                (COND
                   ((WINDOWP PWINDOW)                           ; We found a window to use.  Print the message.
                    (CL:WHEN CLEAR? (CLEARW PWINDOW))
                    (CL:WHEN FLASH? (FLASHWINDOW PWINDOW 1 75))
                    (PRIN1 MSG PWINDOW))
                   (T                                           ; Failing all else, use global PROMPTWINDOW.
                      (FRESHLINE PROMPTWINDOW)
                      (CL:WHEN FLASH? (FLASHWINDOW PWINDOW 1 75))
                      (printout PROMPTWINDOW MSG)))
          else (PROMPTPRINT MSG])
```

(TEDIT.PROMPTCLEAR

```
[LAMBDA (TEXTSTREAM FONT)                                       ; Edited 14-Mar-98 12:52 by rmk:
                                                                ; Edited 14-Oct-87 15:35 by bvm:
```

;; Clears the promptwindow attached to TEXTSTREAM and shrinks it back to a single line in font FONT (or TEDIT.PROMPT.FONT) if it has grown.
;; TEXTSTREAM could actually be a stream on the promptwindow itself.

```
(LET [MW (PW (IF (CAR (NLSETQ (GETPROMPTWINDOW (\TEDIT.MAINW TEXTSTREAM)
                                               NIL NIL T)))
                 ELSEIF (WINDOWPROP (WFROMDS TEXTSTREAM)
                                    'TEDIT.PROMPTWINDOW)
                   THEN (WFROMDS TEXTSTREAM]
     (CL:WHEN PW
         (WINDOWPROP PW 'TEDIT.NLINES 1)
         (CL:WHEN [AND (SETQ MW (WINDOWPROP PW 'MAINWINDOW))
                       (SETQ MW (LISTP (WINDOWPROP MW 'PROMPTWINDOW]
                  (RPLACD MW 1))
         (LET [PROP [HEIGHT (HEIGHTIFWINDOW (FONTPROP (OR FONT TEDIT.PROMPT.FONT)
                                                      'HEIGHT]
                            (REG (WINDOWPROP PW 'REGION]
              (CL:UNLESS (EQ HEIGHT (FETCH HEIGHT OF REG))
                  (WINDOWPROP PW 'MINSIZE (CONS 0 HEIGHT))
                  ;; Have to adjust the fixed size of the window before shaping, since SHAPEW obeys the minimum.
                  (WINDOWPROP PW 'MAXSIZE (CONS 64000 HEIGHT))
                  (SHAPEW PW (CREATE REGION USING REG HEIGHT _ HEIGHT)))
              (CL:WHEN (OPENWP PW)
                     (CLEARW PW)))]])
```

(TEDIT.PROMPTFLASH

```
[LAMBDA (TEXTSTREAM)                                            ; Edited 15-Mar-2024 18:32 by rmk
                                                                ; Edited 30-May-91 23:34 by jds
                                                                ; Flash the TEdit prompt window, or the global promptwindow, if
                                                                ; TEdit has none.
   (PROG (WINDOW PWINDOW (TEXTOBJ (TEXTOBJ TEXTSTREAM))
                MAINTEXTOBJ)
         (COND
            [(AND TEXTOBJ (fetch (TEXTOBJ MENUFLG) of TEXTOBJ))  ; There is a known textobj, and it's a menu.  Go use the main
                                                                ; editor's promptwindow.
             (SETQ MAINTEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of (\TEDIT.MAINW TEXTOBJ)))
                                                                ; Find the TEXTOBJ for the main edit window, and use ITS
                                                                ; prompting window.
             (SETQ WINDOW (AND MAINTEXTOBJ (fetch (TEXTOBJ PROMPTWINDOW) of MAINTEXTOBJ]
            ((AND TEXTOBJ (SETQ WINDOW (fetch (TEXTOBJ PROMPTWINDOW) of TEXTOBJ)))
                                                                ; There IS an editor window to get to;  use its prompt window

             )
            ((SETQ WINDOW (GETPROMPTWINDOW (\TEDIT.MAINW TEXTSTREAM)
                                  NIL NIL T))                   ; Failing that, try any prompt window attached to the edit window.

             ))                                                 ; Try to find an editor's prompt window for our message
         (FLASHWINDOW (OR WINDOW PROMPTWINDOW)
                2])
```

(\TEDIT.PROMPT.PAGEFULLFN

```
[LAMBDA (PROMPT-DISPLAY-STREAM)                                 ; Edited 21-Jun-2024 23:21 by rmk
                                                                ; Edited 18-Nov-87 14:44 by jds
```

;; Given a TEdit promptwindow, expand it to be a line taller--called when a message overflows the window.

```
(LET* [(PROMPT-WINDOW (WFROMDS PROMPT-DISPLAY-STREAM))
       (%#LINES (ADD1 (OR (WINDOWPROP PROMPT-WINDOW 'TEDIT.NLINES)
                          1)))
       (OLDREGION (WINDOWPROP PROMPT-WINDOW 'REGION))
       (OLDTOP (fetch (REGION TOP) of OLDREGION))
       (OLDBOTTOM (fetch (REGION BOTTOM) of OLDREGION))
       (MAINWINDOW (WINDOWPROP PROMPT-WINDOW 'MAINWINDOW))
       (ATTACHEDMENUS (REMOVE PROMPT-WINDOW (ATTACHEDWINDOWS MAINWINDOW]
      (GETPROMPTWINDOW MAINWINDOW %#LINES)                      ; Get the new window
      (SETQ \CURRENTDISPLAYLINE (CL:1- %#LINES))                ; Set this so the page-full code will fire again at the end of THIS
                                                                ; line, rather than waiting for another screen-ful.  There ought to
                                                                ; be an interface to this.
      [for WINDOW [NEWTOP _ (fetch (REGION TOP) of (WINDOWPROP PROMPT-WINDOW 'REGION]
         in (REVERSE ATTACHEDMENUS) when (>= (fetch (REGION BOTTOM) of (WINDOWPROP WINDOW 'REGION))
```

```
                                                OLDBOTTOM)
              do (RELMOVEW WINDOW (CREATEPOSITION 0 (IDIFFERENCE NEWTOP OLDTOP]
           (WINDOWPROP PROMPT-WINDOW 'TEDIT.NLINES %#LINES])

)

(RPAQ? TEDIT.PROMPT.FONT (FONTCREATE 'TERMINAL 10))

(RPAQ? TEDIT.PROMPTWINDOW.HEIGHT NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.PROMPT.FONT TEDIT.PROMPTWINDOW.HEIGHT)
)
```

;; Title creation and update

```
(DEFINEQ
```

(\TEXTSTREAM.TITLE

```
  [LAMBDA (STREAM)                                                    ; Edited 18-Oct-2023 00:02 by rmk
                                                                      ; Edited 24-Aug-2021 23:25 by rmk:
```

;; returns a string with which you can talk to the user about this stream. e.g. for Get and Put prompts

```
    (LET ((TEXTOBJ (TEXTOBJ STREAM))
          TXTFILE)
         (SETQ TXTFILE (FGETTOBJ TEXTOBJ TXTFILE))
         (OR (CL:TYPECASE TXTFILE
                 (STRINGP TXTFILE)
                 (STREAM (fetch (STREAM FULLNAME) of TXTFILE))
                 (LITATOM TXTFILE)
                 (T TXTFILE))
             ""])
```

(\TEDIT.DEFAULT.TITLE

```
  [LAMBDA (FILE PROPS)                                                ; Edited 18-Oct-2023 13:47 by rmk
                                                                      ; Edited 21-Sep-2023 22:47 by rmk
                                                                      ; Edited 17-Sep-2023 09:20 by rmk
                                                                      ; Edited  8-Sep-2023 00:38 by rmk
                                                                      ; Edited 27-Oct-2021 12:25 by rmk:
                                                                      ; Edited 24-Aug-2021 23:25 by rmk:
```

;; Given a file name, derive a title for the TEdit window that is editing it.

```
    (LET [(TITLE (LISTGET PROPS 'TITLE]
         (CL:UNLESS TITLE
             [SETQ TITLE (CONCAT (CL:IF (LISTGET PROPS 'READONLY)
                                        "See "
                                        "Tedit ")
                                (COND
                                   ((NULL FILE)                       ; Just calling (TEDIT) should give a 'Text Editor Window'
                                    "Window")
                                   ((AND (STRINGP FILE)
                                         (ZEROP (NCHARS FILE)))       ; So should editing an empty string
                                    "Window")
                                   ((WINDOWP FILE)                    ; if \TEDIT.WINDOW.SETUP has assigned a title, use it
                                    (OR (WINDOWPROP FILE 'TITLE)
                                        "Window"))
                                   (T                                 ; Strings use the string itself, otherwise grab the full file name.
                                      (CONCAT (CL:TYPECASE FILE
                                                  (STRINGP FILE)
                                                  (STREAM (fetch (STREAM FULLNAME) of FILE))
                                                  (LITATOM FILE)
                                                  (T FILE))])
         TITLE])
```

(\TEDIT.WINDOW.TITLE

```
  [LAMBDA (TEXTOBJ DIRTYFLAG TITLE)                                   ; Edited 25-Jun-2024 11:59 by rmk
                                                                      ; Edited  2-Dec-2023 16:41 by rmk
                                                                      ; Edited 21-Oct-2023 15:02 by rmk
                                                                      ; Edited 18-Oct-2023 00:44 by rmk
                                                                      ; Edited 22-Sep-2023 19:51 by rmk
                                                                      ; Edited 19-Sep-2023 00:47 by rmk
                                                                      ; Edited 24-Oct-2022 13:14 by rmk
                                                                      (* jds "23-May-85 15:20")
```

;; This puts * or clears * in the title of a tedit window. TITLE may override the current window title (e.g. for get and put)

```
    (CL:UNLESS (GETTOBJ TEXTOBJ MENUFLG)
        (LET ((W (\TEDIT.PRIMARYPANE TEXTOBJ)))
             (CL:WHEN (AND W (NOT (GETTEXTPROP TEXTOBJ 'NOTITLE))
                           (WINDOWPROP W 'TEDITCREATED))             ; Only change the title if there IS a window, and it isn't
                                                                     ; suppressing title changes.
                 (if TITLE
                     then (WINDOWPROP W 'TITLE TITLE)
                   else (SETQ TITLE (OR (WINDOWPROP W 'TITLE)
                                        ""])))
```

```
                (CL:UNLESS (EQ DIRTYFLAG (FGETTOBJ TEXTOBJ \XDIRTY))
                    (if DIRTYFLAG
                        then (SETQ TITLE (CONCAT "* " TITLE))
                      elseif (AND (EQ (CHARCODE *)
                                      (CHCON1 TITLE))
                                  (EQ (CHARCODE SPACE)
                                      (NTHCHARCODE TITLE 2)))
                        then (SETQ TITLE (OR (SUBSTRING TITLE 3)
                                             "")))
                    (WINDOWPROP W 'TITLE TITLE))
                TITLE)))])
```

(**TEXTSTREAM.FILENAME**

```
  [LAMBDA (TEXTSTREAM UNFORMATTED?)                                  ; Edited 18-Jan-2024 09:03 by rmk
                                                                     ; Edited 29-Dec-2023 00:33 by rmk
                                                                     ; Edited 18-Dec-2023 14:06 by rmk
                                                                     ; Edited 30-May-91 23:34 by jds
    ;; Offer TXT for plaintext, TEDIT for formatted (including BRAVO)

    ;; If the input file is a foreign format (e.g. bravo), we probably don't want to put out a Tedit or plaintext file with its old extension.  Perhaps the input
    ;; format should mark the stream so as to avoid overwriting.

    ;; returns the name of the file associated with this stream if there is one.  NIL otherwise.  Version numbers suppressed.

    (LET* ((TEXTOBJ (TEXTOBJ TEXTSTREAM))
           (DEFAULTEXT (CL:IF UNFORMATTED?
                           'TXT
                           'TEDIT))
           (TXTFILE (GETTOBJ TEXTOBJ TXTFILE))
           EXT)
          (CL:WHEN (type? STREAM TXTFILE)
              (SETQ TXTFILE (fetch FULLFILENAME of TXTFILE))
              [SETQ EXT (U-CASE (FILENAMEFIELD TXTFILE 'EXTENSION]
              (if (OR (NULL EXT)
                      (EQ EXT 'BRAVO))
                  then (SETQ EXT DEFAULTEXT)
                elseif (AND UNFORMATTED? (MEMB EXT *TEDIT-EXTENSIONS*)
                            (NEQ EXT 'TEXT))
                  then (SETQ EXT 'TXT))
              (PACKFILENAME 'EXTENSION EXT 'VERSION NIL 'BODY TXTFILE))])
```

(**TEDIT.UPDATE.TITLE**

```
  [LAMBDA (TEXTOBJ FILENAME)                                         ; Edited 13-Dec-2024 08:59 by rmk
                                                                     ; Edited 22-Oct-2024 11:44 by rmk
                                                                     ; Edited 28-Aug-2024 15:50 by rmk
                                                                     ; Edited 11-Aug-2024 13:11 by rmk
                                                                     ; Edited 20-Dec-2023 23:44 by rmk
                                                                     ; Edited 18-Oct-2023 09:56 by rmk
                                                                     ; Edited  1-Sep-2023 23:55 by rmk
    ;; find and set the title to reflect a new filename, and update the file fields of any attached menu too.

    (LET ((TITLE (\TEXTSTREAM.TITLE TEXTOBJ))
          MENUSTREAM PC STATEFN)
         (\TEDIT.WINDOW.TITLE TEXTOBJ NIL (\TEDIT.DEFAULT.TITLE (OR FILENAME TITLE)))
         (SETQ MENUSTREAM (TEDITMENU.STREAM TEXTOBJ))
         (CL:WHEN (AND MENUSTREAM (LITATOM TITLE))                   ; if we have a filename then put it in the GET and PUT fields of
                                                                     ; the menu
             (SETQ FILENAME (PACKFILENAME 'VERSION NIL 'BODY TITLE))
             (for BUTTON SETSTATEFN in (MB.GET '(GET PUT)
                                               MENUSTREAM
                                               '(OBJECT STARTPC))
                when (SETQ SETSTATEFN (IMAGEOBJPROP (CAR BUTTON)
                                                    'SETSTATEFN))
                do (APPLY* SETSTATEFN (CADR BUTTON)
                           FILENAME MENUSTREAM)))])

)

;; Screen updating utilities

(DEFINEQ

(TEDIT.DEACTIVATE.WINDOW
  [LAMBDA (PANE)                                                     ; Edited 29-Nov-2024 13:10 by rmk
                                                                     ; Edited  1-Jul-2024 17:42 by rmk
                                                                     ; Edited 18-May-2024 16:20 by rmk
                                                                     ; Edited 12-May-2024 17:19 by rmk
                                                                     ; Edited 15-Mar-2024 13:34 by rmk
                                                                     ; Edited 20-Mar-2024 11:02 by rmk
                                                                     ; Edited 17-Oct-2023 08:54 by rmk
                                                                     ; Edited 10-Oct-2023 10:23 by rmk
                                                                     ; Edited 30-Sep-2023 13:42 by rmk
                                                                     ; Edited 22-Sep-2023 00:07 by rmk
                                                                     ; Edited  9-Mar-2023 15:12 by rmk
                                                                     ; Edited  5-Nov-2022 23:29 by rmk
                                                                     ; Edited 16-Oct-2021 18:51 by rmk:
```

```
;; If the session is or can be finished, deactivate this Tedit window and process, and all attached Tedit menus.  This disconnects the window and
;; process from the textstream, which persists.  This is not used to unsplit panes.  The actual window-closing is done by setting the flag
;; EDITFINISHEDFLG to T and  giving control to the edit process. The flag causes the command loop to exit.

(PROG* [(TSTREAM (TEXTSTREAM PANE T))
        (TEXTOBJ (AND TSTREAM (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM]
       (CL:UNLESS TEXTOBJ                                    ; Return NIL if not an editing window (rather than error?)
           (RETURN))
       (TEXTOBJ! TEXTOBJ)

 ;; Return DON'T to signal (to CLOSEW) that the window shouldn't be closed.  if previously quit, the window is closed already, and would be
 ;; reopened to reclose it.

       (CL:WHEN (\TEDIT.FINISHEDIT? TSTREAM T)
           (RETURN 'DON'T))
       (CL:WHEN (AND (GETTOBJ TEXTOBJ PROMPTWINDOW)
                     (OPENWP (GETTOBJ TEXTOBJ PROMPTWINDOW)))
           (CLEARW (GETTOBJ TEXTOBJ PROMPTWINDOW)))
       (\TEDIT.SETCARET (TEXTSEL TEXTOBJ)
              PANE TEXTOBJ 'OFF)                             ; Before the window is closed, make SURE that the caret is
                                                             ; down, or the window will reappear.
       (CL:WHEN (AND (\TEDIT.WINDOW.TITLE TEXTOBJ)
                     (OPENWP (GETTOBJ TEXTOBJ PROMPTWINDOW))
                     (OPENWP PANE)
                     (EQ PANE (FGETTOBJ TEXTOBJ PRIMARYPANE)))

            ;; Reset the window's title to a known 'inactive' value, in case somebody else also has the window.

            (\TEDIT.WINDOW.TITLE TEXTOBJ NIL "Edit Window [Inactive]"))
       (for PANE backpanes TEXTOBJ do (\TEDIT.UNSPLITW PANE))
       (SETTOBJ TEXTOBJ PRIMARYPANE NIL)
       (CL:WHEN (type? STREAM (GETTOBJ TEXTOBJ TXTFILE))     ; Close the file that this window was open on.
           (CL:UNLESS (fetch (TEXTWINDOW CLOSINGFILE) of PANE)
               (replace (TEXTWINDOW CLOSINGFILE) of PANE with T)
               (CLOSEF? (GETTOBJ TEXTOBJ TXTFILE))))
       (WINDOWPROP PANE 'PROCESS.EXITFN NIL)
       (WINDOWPROP PANE 'PROCESS.IDLEFN NIL)
       (WINDOWPROP PANE 'BUTTONEVENTFN (FUNCTION TOTOPW))    ; And the button functions
       (WINDOWPROP PANE 'RIGHTBUTTONFN (FUNCTION DOWINDOWCOM))
       (WINDOWDELPROP PANE 'CLOSEFN (FUNCTION TEDIT.DEACTIVATE.WINDOW))
                                                             ; To avoid a loop
       (WINDOWPROP PANE 'SCROLLFN NIL)
       (WINDOWDELPROP PANE 'RESHAPEFN (FUNCTION \TEDIT.RESHAPEFN))
       (\TEDIT.INTERRUPT.SETUP (WINDOWPROP PANE 'PROCESS)
              T)                                             ; Restore any disarmed interrupts.
       (for MENUW in (ATTACHEDWINDOWS PANE) when (TEDITMENUP MENUW) do
                                                             ; Detach all the TEDITMENU windows.
                                                                      (SETTOBJ (TEXTOBJ MENUW)
                                                                             EDITFINISHEDFLG T)
                                                             ; Mark it finished so it closes itself
                                                                      (WINDOWPROP MENUW 'TEDITMENU NIL)
                                                             ; And mark it no longer a menu window
                                                                      (GIVE.TTY.PROCESS MENUW)
                                                             ; Then give it a chance to kill itself off
                                                                      (DISMISS 300))
                                                             ; This closes up the other menus
       (GIVE.TTY.PROCESS PANE)                               ; Now kill this one
       (DISMISS 300)
       (WINDOWPROP PANE 'CURSOROUTFN NIL)
       (WINDOWPROP PANE 'CURSORMOVEDFN NIL)
       (\TEDIT.UNLINKPANE PANE)                              ; Disconnect
       (replace (TEXTWINDOW WTEXTSTREAM) of PANE with NIL])
```

(\TEDIT.RESHAPEFN

```
  [LAMBDA (PANE BITS OLDREGION)                              ; Edited 30-Nov-2024 13:30 by rmk
                                                             ; Edited  4-Nov-2024 17:44 by rmk
                                                             ; Edited  6-Jul-2024 17:00 by rmk
                                                             ; Edited 28-Jun-2024 15:14 by rmk
    ;; This tries to display the current top line at the same position relative to the top of PANE.
                                                             ; Edited 25-Jun-2024 15:53 by rmk
    (LET* ((TEXTOBJ (GETTSTR (fetch (TEXTWINDOW WTEXTSTREAM) of PANE)
                             TEXTOBJ))
           (PREG (DSPCLIPPINGREGION NIL PANE))
           (PANEPREFIX (PANEPREFIX PANE))
           (PANEPROPS (PANEPROPS PANE)))

          ;;  Horizontal parameters are common to all panes.  The vertical parameters are not common.

          (WITH PANEPROPS PANEPROPS (SETQ PANEHEIGHT (fetch (REGION HEIGHT) of PREG))
                (SETQ PANEWIDTH (fetch (REGION WIDTH) of PREG))
                (SETQ PANELEFT (fetch (REGION LEFT) of PREG))
                (SETQ PANERIGHT (fetch (REGION RIGHT) of PREG))
                (SETQ PANEBOTTOM (fetch (REGION BOTTOM) of PREG))
                (SETQ PANETOP (fetch (REGION TOP) of PREG))
                (SETQ PANEREGION PREG))
          (WITH TEXTOBJ TEXTOBJ (SETQ WRIGHT (fetch (REGION WIDTH) of PREG))
                (SETQ WLEFT (fetch (REGION LEFT) of PREG))
                (SETQ WBOTTOM (fetch (REGION BOTTOM) of PREG)))
          [SETYBOT PANEPREFIX (IPLUS (FGETLD PANEPREFIX YBOT)
```

```
                                    (IDIFFERENCE (PANEHEIGHT PANE)
                                           (fetch (REGION HEIGHT) of OLDREGION]
          (CL:WHEN (PANETOPLINE PANE)
              (SETYTOP (PANETOPLINE PANE)
                       (FGETLD PANEPREFIX YBOT)))
          (\TEDIT.FILL.PANES PANE])
```

(\TEDIT.REPAINTFN

```
  [LAMBDA (WINDOW REGION)                                    ; Edited 26-Oct-2024 11:12 by rmk
    ;; Ignores REGION, repaints all the panes
    (\TEDIT.FILL.PANES WINDOW])

)

(DEFINEQ
```

(\TEDIT.SCROLLFN

```
  [LAMBDA (PANE DX DY)
    ;; Edited 29-Apr-2024 15:04 by rmk
    ;; Edited 27-Apr-2024 11:31 by rmk
    ;; Edited 24-Apr-2024 11:28 by rmk
    ;; Edited 20-Mar-2024 11:02 by rmk
    ;; Edited 10-Mar-2024 22:23 by rmk
    ;; Edited 22-Oct-2023 08:31 by rmk
    ;; Edited 11-May-2023 12:03 by rmk
    ;; Edited 18-Feb-2022 14:53 by rmk: Repaint after scrolling for panes that are partially off-screen
    (TOTOPW PANE)
    (PROG* [(TSTREAM (fetch (TEXTWINDOW WTEXTSTREAM) of PANE))
            (TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM]
           (if (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
               then ;; Don't scroll a zero-length file
                    (RETURN)
             elseif (FGETTOBJ TEXTOBJ EDITOPACTIVE)
               then ;; Don't scroll while something interesting is happening!
                    (\TEDIT.PROMPTPRINT TEXTOBJ "Edit operation in progress." T)
                    (RETURN))
           (CL:UNLESS (\GETSTREAM TSTREAM 'INPUT T)
                  (\TEDIT.REOPENTEXTSTREAM TEXTOBJ))
           (CL:WHEN (GETTEXTPROP TEXTOBJ 'PRESCROLLFN)
               (DOUSERFNS (GETTEXTPROP TEXTOBJ 'PRESCROLLFN)
                      PANE))                                 ; Turn off selections during the scroll.
           (if (FLOATP DY)
               then (\TEDIT.SCROLLCH.TOP TSTREAM PANE (IMAX [IMIN (SUB1 (TEXTLEN TEXTOBJ))
                                                                  (FIXR (FTIMES DY (TEXTLEN TEXTOBJ]
                                                          1))
             elseif (IGREATERP DY 0)
               then (\TEDIT.SCROLLUP TSTREAM PANE DY)
             elseif (ILESSP DY 0)
               then (\TEDIT.SCROLLDOWN TSTREAM PANE DY))
           (CL:WHEN (GETTEXTPROP TEXTOBJ 'POSTSCROLLFN)      ; For user subsystem cleanup
               (DOUSERFNS (GETTEXTPROP TEXTOBJ 'POSTSCROLLFN)
                      PANE)))
    NIL])
```

(\TEDIT.SCROLLCH.TOP

```
  [LAMBDA (TSTREAM PANE CHNO)                                ; Edited 17-Nov-2024 14:05 by rmk
                                                             ; Edited 10-Nov-2024 11:54 by rmk
                                                             ; Edited  2-Nov-2024 23:34 by rmk
                                                             ; Edited 31-Oct-2024 14:35 by rmk
                                                             ; Edited  2-Oct-2024 23:55 by rmk
                                                             ; Edited 28-Jun-2024 15:16 by rmk
                                                             ; Edited 18-May-2024 16:20 by rmk
                                                             ; Edited 25-Apr-2024 11:08 by rmk
    ;; Scrolls so that the line containing CHNO is at the top of PANE. This is the body of the earlier \TEDIT.SCROLLFLOAT. This is called for an explicit
    ;; FLOATP scroll or normalize caret.
    (LET ((TEXTOBJ (TEXTOBJ! (GETTSTR TSTREAM TEXTOBJ)))
          (PANEPREFIX (PANEPREFIX PANE))
          TARGETLINE)
         (CL:UNLESS (AND (PANETOPLINE PANE)
                         (WITHINLINEP CHNO (PANETOPLINE PANE)))
             (if (SETQ TARGETLINE (\TEDIT.VISIBLECHARP CHNO PANE TEXTOBJ))
                 then ;; A valid line containg CHNO is on-screen, scroll it to the top of PANE.
                      (\TEDIT.SCROLLUP TSTREAM PANE TARGETLINE)
               elseif (SETQ TARGETLINE (CDR (\TEDIT.LINES.ABOVE TSTREAM CHNO)))
                 then ;; No existing CHNO line.  Construct one and install it as PANE's top line.
```

```
                    (\TEDIT.SETPANE.TOPLINE PANE TARGETLINE (\TEDIT.TOPLINE.YTOP TARGETLINE NIL PANE TEXTOBJ))
                    (\TEDIT.FILL.PANES TEXTOBJ PANE))))])
```

(\TEDIT.SCROLLCH.BOTTOM

```
[LAMBDA (TSTREAM PANE CHNO BOTMARGIN)                                  ; Edited  1-Dec-2024 11:26 by rmk
                                                                       ; Edited 29-Nov-2024 09:14 by rmk
                                                                       ; Edited 21-Nov-2024 10:40 by rmk
                                                                       ; Edited 20-Nov-2024 00:37 by rmk
                                                                       ; Edited 11-Nov-2024 22:40 by rmk
                                                                       ; Edited 10-Nov-2024 12:27 by rmk
                                                                       ; Edited 26-Oct-2024 11:05 by rmk
                                                                       ; Edited 28-Jun-2024 15:15 by rmk
                                                                       ; Edited 21-Jun-2024 23:17 by rmk
                                                                       ; Edited 21-May-2024 15:40 by rmk
                                                                       ; Edited 16-May-2024 23:42 by rmk
                                                                       ; Edited  2-May-2024 00:27 by rmk
                                                                       ; Edited 29-Apr-2024 11:18 by rmk
                                                                       ; Edited 25-Apr-2024 11:08 by rmk
```

;; Scrolls so that the line containing CHNO is at the bottom of PANE. Presumably this is only called when typing at the bottom of a pane, doesn't ;; have to be efficient.

```
(LET ((TEXTOBJ (TEXTOBJ! (GETTSTR TSTREAM TEXTOBJ)))
      (BOTTOM (PANEBOTTOM PANE))
      TARGETLINE TOPLINE NEWLINES PANETOPLINE)
     (CL:WHEN BOTMARGIN (add BOTTOM BOTMARGIN))
```

;; 0 if PANE starts at the beginning of the stream, can't scroll down. If it's current visible, we scan scroll down to its new position. If not ;; visible, we have to search for a new top line somewhere above the CHNO target line such that putting that line at the to will cause the ;; target line to appear at the bottom.

```
     (if (SETQ TARGETLINE (\TEDIT.VISIBLECHARP CHNO PANE TEXTOBJ))
         then ;; CHNO currently visible
              (CL:WHEN (OR (FGETLD TARGETLINE NEXTLINE)
                           (IGREATERP (FGETLD TARGETLINE LCHARLIM)
                                      (FGETTOBJ TEXTOBJ TEXTLEN)))
                  ;; Don't scroll if TARGETLINE is already PANE's last line, unless it's the last line of the document.
                  (\TEDIT.SCROLLDOWN TSTREAM PANE (IDIFFERENCE BOTTOM (FGETLD TARGETLINE YTOP))))
       else ;; CHNO is not in PANE.  Create a line that is PHEIGHT above a line containing CHNO--that's the new top. We want to make sure
            ;; that the bottom of the line containing CHNO is visible
            (SETQ NEWLINES (\TEDIT.BACKFORMAT TSTREAM (PANEHEIGHT PANE)
                                              CHNO BOTTOM))
            ;; TARGETLINE with CHNO is at bottom if TOPLINE is at top
            (SETQ TARGETLINE (CDR NEWLINES))
            (SETQ TOPLINE (CAR NEWLINES))
            ;; Special case where TARGET line is already at the top but it's bottom is not visible.  Scroll up just a bit.  (Presumably this is the
            ;; tall-object case).
            (if (AND (EQ TARGETLINE TOPLINE)
                     (SETQ PANETOPLINE (PANETOPLINE PANE))
                     (EQ (FGETLD TARGETLINE LCHAR1)
                         (FGETLD PANETOPLINE LCHAR1))
                     (ILESSP (FGETLD PANETOPLINE YBOT)
                             BOTTOM))
                then (\TEDIT.SCROLLUP TSTREAM PANE (IDIFFERENCE BOTTOM (FGETLD PANETOPLINE YBOT)))
              else (CL:WHEN (FGETLD TOPLINE NEXTLINE)          ; Lift one line down
                        (SETQ TOPLINE (FGETLD TOPLINE NEXTLINE)))
                   (\TEDIT.SCROLLCH.TOP TSTREAM PANE (FGETLD TOPLINE LCHARLAST])
```

(\TEDIT.SCROLLUP

```
[LAMBDA (TSTREAM PANE DY)                                              ; Edited  1-Feb-2025 10:20 by rmk
                                                                       ; Edited  1-Dec-2024 11:32 by rmk
                                                                       ; Edited 29-Nov-2024 09:14 by rmk
                                                                       ; Edited 22-Nov-2024 17:33 by rmk
                                                                       ; Edited 21-Nov-2024 15:04 by rmk
                                                                       ; Edited 20-Nov-2024 12:37 by rmk
                                                                       ; Edited 17-Nov-2024 19:00 by rmk
                                                                       ; Edited 15-Nov-2024 20:19 by rmk
                                                                       ; Edited 11-Nov-2024 22:26 by rmk
                                                                       ; Edited  9-Nov-2024 23:26 by rmk
                                                                       ; Edited  7-Nov-2024 16:58 by rmk
                                                                       ; Edited  2-Nov-2024 19:00 by rmk
                                                                       ; Edited 26-Oct-2024 15:55 by rmk
                                                                       ; Edited 28-Jun-2024 15:18 by rmk
                                                                       ; Edited  9-May-2024 07:43 by rmk
                                                                       ; Edited 20-Mar-2024 06:43 by rmk
                                                                       ; Edited 14-Dec-2023 00:00 by rmk
                                                                       ; Edited 24-Apr-2023 23:48 by rmk
```

;; Scrolling up, with positive integer DY. We find a line that is or would be DY below the top of the pane, then move that line to the top and fill in ;; beneath.

```
(PROG ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
       (OLDTOPLINE (PANETOPLINE PANE))
```

```
          NEWTOPLINE NEWPANEYBOT BITMAPLINES)
    ;; Find the first line at least DY below the top of the pane.
        (CL:UNLESS OLDTOPLINE
            ;; Relative scrolling doesn't make sense if there isn't at least one visible line currently at the top of the pane.
            (RETURN))
    ;;
    ;; Walk down the sequence of lines until we arrive at a line whose top is at least DY below the top of PANE. If we run off the bottom of existing
    ;; lines, keep formatting until we finally exhaust DY or reach the end of the text. Unlike the scroll-down case, we know we are starting from a
    ;; properly broken line, we don't have to search for a stable paragraph break.
        [SETQ NEWTOPLINE (if (type? LINEDESCRIPTOR DY)
                             then (PROG1 DY (SETQ DY NIL))
                           elseif (IGREATERP DY 0)
                             then (for L NEXT TARGETY (SUFFIX _ (PANESUFFIX PANE))
                                       (PHEIGHT _ (PANEHEIGHT PANE))
                                       (TEXTLEN _ (TEXTLEN TEXTOBJ))
                                       inlines OLDTOPLINE first (SETQ TARGETY (IDIFFERENCE PHEIGHT DY))
                                    do (CL:WHEN (OR (\TEDIT.SHOW.AT.TOPP L TARGETY PHEIGHT)
                                                    (IGREATERP (FGETLD L LCHARLIM)
                                                               TEXTLEN))
                                               (RETURN L))
                                       (CL:WHEN (EQ L SUFFIX)
                                            ;; Continue by formatting a new, undisplayed line.  This can happen if DY (say from an explicit
                                            ;; SCROLLW call) picks a line that is somewhere below the pane.  The newline is linked in as L's
                                            ;; NEXTLINE, but its NEXTLINE is NIL, so we keep running through here. The new lines are
                                            ;; positioned propely with respect to the current OLDTOPLINE (so are all lines we have crossed over
                                            ;; or formatted, but those will be thrown away.)
                                            (SETQ NEXT (\TEDIT.FORMATLINE TSTREAM (FGETLD L LCHARLIM)))
                                            (LINKLD L NEXT)     ; So we find NEXT on the next iteration
                                            (SETYBOT NEXT (\TEDIT.LINE.BOTTOM L NEXT)))]
        (CL:UNLESS NEWTOPLINE                                   ; If nothing found, nothing can be done
            (RETURN))
    ;;
        (CL:UNLESS (SETQ NEWPANEYBOT (\TEDIT.TOPLINE.YTOP NEWTOPLINE DY PANE TEXTOBJ))
               (RETURN NIL))
    ;; Position and display lines such that NEWTOPLINE's top is at YBOT of the pane prefix.
        (\TEDIT.SETPANE.TOPLINE PANE NEWTOPLINE NEWPANEYBOT)
        (\TEDIT.SHIFTLINES (PANEPREFIX PANE)
               PANE TEXTOBJ (\TEDIT.BITMAPLINES PANE NEWTOPLINE)
               T)
        (\TEDIT.SETCARET (TEXTSEL TEXTOBJ)
               PANE TEXTOBJ 'ON])
```

(\TEDIT.TOPLINE.YTOP

```
  [LAMBDA (NEWTOPLINE DY PANE TEXTOB)                       ; Edited 29-Nov-2024 09:14 by rmk
                                                            ; Edited 11-Nov-2024 22:30 by rmk
                                                            ; Edited  9-Nov-2024 23:51 by rmk
    ;; Return a value of YTOP value for NEWTOPLINE such that at least part of it will be visible at the top of PANE if it is shifted up by DY points
    ;; (unless DY is NIL).  NEWTOPLINE is already known to be the plus-DY target line and it is already known that at least some part of it can be
    ;; displayed.
    ;; Constraints:
    ;;   1. If the document isn't empty, at least (a piece of) one line should be visible.
    ;;   2. If the line is tall with respect to PANE, then it is OK to show only part of the line.
    ;; NIL is returned if NEWTOPLINE is already at the target level.
    (PROG ((PHEIGHT (PANEHEIGHT PANE))
           NEWTOP NEWBOT)
          (CL:UNLESS (AND DY (\TEDIT.LINE.TALLP NEWTOPLINE PHEIGHT))
                                                            ; Case 1 and 3
              (RETURN (IPLUS PHEIGHT (FGETLD NEWTOPLINE LLEADBEFORE))))
     ;; The tall line maybe coming up from below, or it may already have been at the top.
          (SETQ NEWBOT (IPLUS DY (FGETLD NEWTOPLINE YBOT)))
          (CL:WHEN (IGREATERP NEWBOT PHEIGHT)               ; Case 2: Bring it back down
              (SETQ NEWBOT (IDIFFERENCE PHEIGHT 2)))
          (RETURN (IPLUS NEWBOT (FGETLD NEWTOPLINE LHEIGHT])
```

(\TEDIT.SCROLLDOWN

```
  [LAMBDA (TSTREAM PANE DY)                                 ; Edited  1-Feb-2025 10:20 by rmk
                                                            ; Edited  1-Dec-2024 20:46 by rmk
                                                            ; Edited 29-Nov-2024 09:14 by rmk
                                                            ; Edited 22-Nov-2024 17:33 by rmk
                                                            ; Edited 17-Nov-2024 10:13 by rmk
                                                            ; Edited 15-Nov-2024 19:55 by rmk
                                                            ; Edited 11-Nov-2024 23:58 by rmk
                                                            ; Edited  7-Nov-2024 11:59 by rmk
                                                            ; Edited  1-Nov-2024 15:40 by rmk
                                                            ; Edited 28-Jun-2024 15:19 by rmk
```

```
                                                                    ; Edited 29-Apr-2024 15:06 by rmk
                                                                    ; Edited 19-Mar-2024 23:34 by rmk
                                                                    ; Edited 20-Jan-2024 23:13 by rmk
                                                                    ; Edited  1-Dec-2023 16:11 by rmk
                                                                    ; Edited 11-May-2023 11:53 by rmk
                                                                    ; Edited 26-Mar-2023 20:55 by rmk
```

;; Add new lines that fill DYat the top of PANE. The needed lines are first constructed, then pushed in front of any old lines. The NEWTOPLINE is ;; positioned at the top of the window, and all other lines are then positioned relative to it. The current Y positions of all lines are ignored, new ;; positions are determined based on LHEIGHTs.

;; The bitmap corresponding to the old-line positions are moved so that they correlate with their new positions, resulting garbage at the bottom of ;; the window is cleared, and then the newlines are displayed to fill the space at the top.

;;

```
(PROG* ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
        (PBOTTOM (PANEBOTTOM PANE))
        (PHEIGHT (PANEHEIGHT PANE))
        (OLDTOPLINE (PANETOPLINE PANE))
       NEWLINES NEWTOPLINE NEWPANEYBOT)
      (CL:UNLESS (OR OLDTOPLINE (EQ 0 (TEXTLEN TEXTOBJ)))
             (RETURN))
```

;; Look backwards from the line before OLDTOP.

```
      [if (type? LINEDESCRIPTOR DY)
          then (SETQ NEWTOPLINE DY)
               (SETQ DY 0)
        else (CL:WHEN (EQ DY 0)
                  (RETURN))
             (SETQ DY (IMINUS DY))                          ; Now positive
             (if (AND (IGREATERP (IDIFFERENCE (IDIFFERENCE (FGETLD OLDTOPLINE YTOP)
                                                           DY)
                                              PHEIGHT)
                                 (FGETLD OLDTOPLINE LLEADBEFORE))
                      (\TEDIT.LINE.TALLP OLDTOPLINE PHEIGHT))
                 then
```

;; OLDTOPLINE must have a big image object. If it's top is still above the window when it is brought down by DY, ;; then just shift the image here. Otherwise, code below will bring down the line above.

```
                      (SETQ NEWTOPLINE OLDTOPLINE)
```

;; We want to show DY more at the bottom of the large line

```
                      (SETQ NEWPANEYBOT (IDIFFERENCE (FGETLD NEWTOPLINE YTOP)
                                                     DY))
                 else (SETQ NEWLINES (\TEDIT.BACKFORMAT TSTREAM DY (IMAX 1 (SUB1 (FGETLD OLDTOPLINE LCHAR1)))
                                                        0))
                      (CL:UNLESS NEWLINES                   ; Should always find one, but...
                          (RETURN))
                      (SETQ NEWTOPLINE (CAR NEWLINES))
                      (CL:WHEN (IEQP (FGETLD NEWTOPLINE LCHAR1)
                                     (FGETLD OLDTOPLINE LCHAR1))
                                                            ; BACKFORMAT can produce a copy
                          (SETQ NEWTOPLINE OLDTOPLINE))
                      (CL:WHEN [AND (EQ 1 (FGETLD OLDTOPLINE LCHAR1))
                                    (OR (ILEQ (FGETLD OLDTOPLINE LTRUEYTOP)
                                              PHEIGHT)
                                        (NOT (\TEDIT.LINE.TALLP OLDTOPLINE PHEIGHT]
```

;; Top line of document, only pull it down if it is tall and had been scrolled up.

```
                          (RETURN))
                      (LINKLD (CDR NEWLINES)
                              OLDTOPLINE)
                      (SETQ NEWPANEYBOT (if (EQ NEWTOPLINE OLDTOPLINE)
                                            then
```

;; Top line that passed the tall test

```
                                                 (IDIFFERENCE (FGETLD OLDTOPLINE LTRUEYTOP)
                                                              DY)
                                            elseif (\TEDIT.LINE.TALLP NEWTOPLINE PHEIGHT)
                                            then
```

;; Try to show just DY at the bottom of the tall line. Maybe the tall-line branches should ;; be collected under a single condition?

```
                                                 (IPLUS (FGETLD NEWTOPLINE LTRUEHEIGHT)
                                                        (IDIFFERENCE PHEIGHT DY))
                                            elseif (IPLUS PHEIGHT (FGETLD NEWTOPLINE LLEADBEFORE]
```

;; NEWTOPLINE is the top of a chain at or above OLDTOPLINE.

;;

```
      (\TEDIT.SETPANE.TOPLINE PANE NEWTOPLINE NEWPANEYBOT)
```

;; All needed lines have been constructed and linked, although there may still be some unneeded lines at the bottom.

```
      (\TEDIT.SHIFTLINES (PANEPREFIX PANE)
             PANE TEXTOBJ (\TEDIT.BITMAPLINES PANE OLDTOPLINE)
             T)
      (\TEDIT.SETCARET (TEXTSEL TEXTOBJ)
             PANE TEXTOBJ 'ON])
```

```
[LAMBDA (TSTREAM)                                                    ; Edited 21-Nov-2024 10:39 by rmk
                                                                     ; Edited 17-Nov-2024 19:00 by rmk
                                                                     ; Edited 11-Nov-2024 22:07 by rmk
                                                                     ; Edited 26-Oct-2024 11:06 by rmk
                                                                     ; Edited  3-Oct-2024 08:53 by rmk
                                                                     ; Edited  1-Jul-2024 18:30 by rmk
                                                                     ; Edited 28-Jun-2024 15:26 by rmk
                                                                     ; Edited 22-Jun-2024 00:10 by rmk
                                                                     ; Edited 21-May-2024 15:40 by rmk
                                                                     ; Edited 16-May-2024 23:42 by rmk
                                                                     ; Edited  2-May-2024 00:27 by rmk
                                                                     ; Edited 29-Apr-2024 11:18 by rmk
                                                                     ; Edited 25-Apr-2024 11:08 by rmk
    ;; Makes sure that the line containing the caret is visible in PANE, assuming that the caret and the current lines both reflect the same character
    ;; numbers. For tall lines we want to be sure that the line's base is visible, that's where the caret flashes..

    ;; If the caret is above the pane, move it down to the top line.

    ;; If the caret is visible in PANE, nothing to do.

    ;; If it is below the pane, then move its line to the bottom.

    (TEDIT.PROMPTCLEAR TSTREAM)
    (PROG ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
           PANE CARETCHNO)
          (SETQ PANE (OR (FGETTOBJ TEXTOBJ SELPANE)
                         (\TEDIT.PRIMARYPANE TSTREAM)))
          (CL:UNLESS PANE (RETURN))
          (CL:WHEN (\TEDIT.VISIBLECARETP PANE TEXTOBJ)
                 (RETURN))
          (SETQ CARETCHNO (TEDIT.GETPOINT TSTREAM))
          (if (ILEQ CARETCHNO (FGETLD (PANEPREFIX PANE)
                                      LCHARLAST))
              then                                                   ; Caret is above PANE
                   (\TEDIT.SCROLLCH.TOP TSTREAM PANE CARETCHNO)
            else ;; CARETCHNO must be below PANE.  Make a line to contain it at the bottom of PANE.
                 (\TEDIT.SCROLLCH.BOTTOM TSTREAM PANE CARETCHNO 3])
```

(\TEDIT.VISIBLECARETP

```
[LAMBDA (PANE TEXTOBJ)                                               ; Edited  1-Dec-2024 11:26 by rmk
                                                                     ; Edited 29-Nov-2024 13:28 by rmk
                                                                     ; Edited 18-Nov-2024 15:49 by rmk
                                                                     ; Edited 17-Nov-2024 14:07 by rmk
                                                                     ; Edited 11-Nov-2024 21:47 by rmk
    ;; Returns the line in PANE that contains the caret, if the caret is currently positioned in PANE, even though it may temporally be suspended.  This
    ;; ssumes that the caret is positioned at the base of its line, and that the caret's base position can be retrieved from the line.  If the line has a
    ;; forced-end, the caret is presumably blinking at the front of the next line, so this returns NIL

    (LET ((CARETCHNO (SUB1 (TEDIT.GETPOINT TEXTOBJ)))
          (PCARET (PANECARET PANE))
          CARETLINE)
         ;; Maybe X has to be in the Text region?
         (CL:WHEN (AND (fetch (TEDITCARET TCCARETY) of PCARET)
                       (fetch (TEDITCARET TCCARETX) of PCARET)
                       (IGEQ (fetch (TEDITCARET TCCARETX) of PCARET)
                             0)
                       (IBETWEENP (fetch (TEDITCARET TCCARETY) of PCARET)
                                  (PANEBOTTOM PANE)
                                  (PANEHEIGHT PANE)))
             (CL:WHEN (AND (IGREATERP CARETCHNO (TEXTLEN TEXTOBJ))
                           (IGREATERP (TEXTLEN TEXTOBJ)
                                      0))
                 (SETQ CARETCHNO (TEXTLEN TEXTOBJ)))
             (SETQ CARETLINE (\TEDIT.VISIBLECHARP CARETCHNO PANE TEXTOBJ))
             (CL:WHEN NIL
                 (AND (FGETLD CARETLINE FORCED-END)
                      (FGETLD CARETLINE NEXTLINE))
                 (SETQ CARETLINE (FGETLD CARETLINE NEXTLINE)))
             (CL:IF (AND CARETLINE (IBETWEENP (FGETLD CARETLINE YBASE)
                                              (PANEBOTTOM PANE)
                                              (PANEHEIGHT PANE)))
                    CARETLINE)])
```

(\TEDIT.VISIBLECHARP

```
[LAMBDA (CHNO PANE TEXTOBJ)                                          ; Edited  1-Dec-2024 11:26 by rmk
                                                                     ; Edited 29-Nov-2024 09:15 by rmk
                                                                     ; Edited 17-Nov-2024 19:01 by rmk
                                                                     ; Edited 11-Nov-2024 16:52 by rmk
                                                                     ; Edited  2-Oct-2024 23:52 by rmk
    ;; Returns a visible line in PANE that contains CHNO, if any.

    (CL:WHEN (IGEQ CHNO (FGETLD (PANEPREFIX PANE)
                                LCHARLIM))
        (find L (TOP _ (PANEHEIGHT PANE))
              (BOTTOM _ (PANEBOTTOM PANE))
```

```
            inlines
            (PANETOPLINE PANE) suchthat (AND (WITHINLINEP CHNO L)
                                             (ILEQ (FGETLD L LTRUEYTOP)
                                                   TOP)
                                             (IGEQ (FGETLD L YBOT)
                                                   BOTTOM))))])
```

(\TEDIT.BITMAPLINES

```
[LAMBDA (PANE STARTLINE ENDLINE)                                    ; Edited  1-Dec-2024 11:28 by rmk
                                                                    ; Edited 29-Nov-2024 22:45 by rmk
                                                                    ; Edited 24-Nov-2024 14:39 by rmk
                                                                    ; Edited 22-Nov-2024 17:08 by rmk
                                                                    ; Edited 19-Nov-2024 23:24 by rmk
                                                                    ; Edited 16-Nov-2024 21:15 by rmk
                                                                    ; Edited 11-Nov-2024 23:43 by rmk
                                                                    ; Edited  8-Nov-2024 17:08 by rmk
                                                                    ; Edited  5-Nov-2024 12:54 by rmk
```

;; This returns the completely visible top and bottom lines of PANE's on-screen bitmap, from STARTLINE (or its next, if it isn't completely visible) to ;; the bottom of PANE.

;; If the top of the top line is above PANE, its bitmap has been clipped. Go to the nextline.

```
(CL:WHEN STARTLINE
    (CL:UNLESS (GETLD STARTLINE LDUMMY)
        (CL:WHEN (AND STARTLINE (IGREATERP (FGETLD STARTLINE YTOP)
                                           (PANEHEIGHT PANE)))
            (SETQ STARTLINE (FGETLD STARTLINE NEXTLINE))
            (CL:WHEN (EQ STARTLINE (PANESUFFIX PANE))
                (SETQ STARTLINE NIL)))
        [for L (PHEIGHT _ (PANEHEIGHT PANE))
            (PBOTTOM _ (PANEBOTTOM PANE))
            (VHEIGHT _ (\TEDIT.ONSCREEN? PANE 'HEIGHT))
            (VBOTTOM _ (\TEDIT.ONSCREEN? PANE 'BOTTOM))
            inlines STARTLINE first
```

;; Visible height and visible bottom are in PANE's coordinate system. The bitmap is only good for ;; onscreen lines, so we have to calculate the onscreen height and onscreen bottom.

;; PANE could be offscreen on the top, bottom, or both.

```
            (if (IEQP VBOTTOM PBOTTOM)
                then ;; Bottom is on screen, VHEIGHT is good
              elseif (IGREATERP PHEIGHT (IPLUS VBOTTOM VHEIGHT))
                then ;; Top is offscreen, VHEIGHT is good
                else ;; Top is onscreen, VBOTTOM shouldn't figure into height
                     (SETQ VHEIGHT PHEIGHT))
```

;; It wouldn't be harmful to let the bitmap include lines at the bottom that are below the screen, if we ;; are moving down. No sense in formatting and displaying those lines. Or include lines above the ;; screen so that they wouldn't be formatted either.

```
            when (ILEQ (FGETLD L LTRUEYTOP)
                       VHEIGHT)
            do (RETURN (CL:WHEN (IGEQ (FGETLD L YBOT)
                                      VBOTTOM)
```

;; If a first line is visible, there must be a last.

```
                       (CONS L (find LL (TOOFAR _ (CL:IF ENDLINE
                                                         (FGETLD ENDLINE NEXTLINE)
                                                         (PANESUFFIX PANE)))
                                  inlines L until (EQ LL TOOFAR) suchthat (ILESSP (FGETLD LL YBOT)
                                                                                  VBOTTOM)
                                  finally (RETURN $$PREVLINE))))]))])
```

(\TEDIT.SETPANE.TOPLINE

```
[LAMBDA (PANE TOPLINE PREFIXYBOT)                                   ; Edited  7-Nov-2024 08:50 by rmk
                                                                    ; Edited  4-Nov-2024 23:05 by rmk
```

;; Install TOPLINE as the PANETOPLINE of PANE, setting PANE's YBOT to PREFIXYBOT if given or the YTOP of TOPLINE. In the ;; PREFIXYBOT case, the pane will be inconsistent until either the line or the pane is adjusted. But before that, the difference between the PANE ;; YBOT and TOPLINE YTOP may be useful.

```
(LET ((PREFIX (PANEPREFIX PANE)))
     (SETLD PREFIX YBOT (OR PREFIXYBOT (FGETLD TOPLINE YTOP)))
     (LINKLD PREFIX TOPLINE)
     (\TEDIT.PREFIX.LCHARLIM PANE (SUB1 (FGETLD TOPLINE LCHAR1)))
     TOPLINE])
```

(\TEDIT.SHIFTLINES

```
[LAMBDA (PREVLINE PANE TEXTOBJ BITMAPLINES SCROLLING)               ; Edited  1-Feb-2025 10:22 by rmk
                                                                    ; Edited  7-Jan-2025 11:54 by rmk
                                                                    ; Edited 17-Dec-2024 23:40 by rmk
                                                                    ; Edited  3-Dec-2024 16:08 by rmk
                                                                    ; Edited  1-Dec-2024 11:31 by rmk
```

; Edited 24-Nov-2024 11:45 by rmk
; Edited 22-Nov-2024 18:00 by rmk
; Edited 19-Nov-2024 23:23 by rmk
; Edited 17-Nov-2024 10:25 by rmk
; Edited 11-Nov-2024 23:51 by rmk

;; BITMAPLINES contains the first and last lines of the currently resuable PANE bitmap. PANE is refilled from the next of PREVLINE to the
;; bottom, using BITMAPLINES and BITBLT to translate the images for lines that are already known. This skips formatting and redisplaying of those
;; lines, but more importantly, it suppresses flicker.

;; Take down the caret, but importantly, don't take down the selection--that would wipe out the bitmap-highlighting that we want to translate.

```
(LET ((SEL (TEXTSEL TEXTOBJ))
      LASTVISIBLE)
     (\TEDIT.SETCARET SEL PANE TEXTOBJ 'OFF)
     (if BITMAPLINES
         then [LET* ((NEXTLINE (FGETLD PREVLINE NEXTLINE))
                     (VLEFT (\TEDIT.ONSCREEN? PANE 'LEFT))
                     (PBOTTOM (PANEBOTTOM PANE))
                     (BMTOPL (CAR BITMAPLINES))
                     (BMTOPY (FGETLD BMTOPL YTOP))
                     (BMBOTL (CDR BITMAPLINES))
                     (BMBOTY (FGETLD BMBOTL YBOT))
                     DELTA)
```

;;; REPOSITION all lines in the chain properly with respect to PREVLINE.

```
                    [for L (Y _ (FGETLD PREVLINE YBOT))
                        inlines NEXTLINE do (SETYTOP L Y)
                                            (SETQ Y (IDIFFERENCE Y (FGETLD L LHEIGHT]
```

;;; TRANSLATE the bitmap to be consistent with its new line positions. This is done before any display operations, to be sure that the bitmap isn't
;;; corrupted.

```
                    (SETQ DELTA (IDIFFERENCE (FGETLD BMTOPL YTOP)
                                             BMTOPY))
                    (BITBLT PANE VLEFT BMBOTY PANE VLEFT (IPLUS BMBOTY DELTA)
                           (PANEWIDTH PANE)
                           (IDIFFERENCE BMTOPY BMBOTY)
                           'INPUT
                           'REPLACE)
                    (SETQ BMTOPY (FGETLD BMTOPL YTOP))
```

;;; Display any lines ABOVE the top of the translated bitmap, presumably for scroll down and insertion. Lines exist and have been formatted and
;;; positioned, but not yet displayed.

```
                    (for L inlines NEXTLINE while (IGEQ (FGETLD L YBOT)
                                                        BMTOPY)
                       do (\TEDIT.DISPLAYLINE TEXTOBJ L PANE))
```

;;; Deal with lines BELOW the bitmap. First. clear to the bottom--important to clear before displaying

```
                    (for L backlines BMBOTL while (AND (ILESSP (FGETLD BMBOTL YBOT)
                                                               PBOTTOM)
                                                       (NOT (\TEDIT.SHOW.AT.BOTTOMP BMBOTL PANE)))
                       do (SETQ BMBOTL (FGETLD BMBOTL PREVLINE)))
                    (\TEDIT.CLEARPANE.BELOW.LINE BMBOTL PANE TEXTOBJ)
                    [SETQ LASTVISIBLE (if (EQ BMBOTL (PANESUFFIX PANE))
                                          then (PANEBOTTOMLINE PANE)
                                        elseif (IGEQ (FGETLD BMBOTL YBOT)
                                                     PBOTTOM)
                                          then ;; Bitmap didn't fill the pane.  Maybe more lines needed below (scroll up or deletion).
                                               (\TEDIT.LINES.BELOW BMBOTL PANE TEXTOBJ)
                                          else ;; Bit map went below the bottom, back up to the previous visible line. (scroll down or
                                               ;; insertion)
                                               (find L backlines BMBOTL suchthat (IGREATERP (FGETLD L YBOT)
                                                                                            PBOTTOM]
                    (\TEDIT.SUFFIXLINE.CREATE PANE TEXTOBJ LASTVISIBLE)
                    ;; Lines are now properly linked, positioned, and displayed.
```

;;;

;;; The part of the current SELECTION within the bitmap retains its correct highlighting, but highlighting has to be applied to lines above or below.

```
                    (\TEDIT.FIXSEL SEL TEXTOBJ NIL PANE)
                    (CL:WHEN (AND (FGETSEL SEL ONFLG)
                                  (NEQ 0 (FGETSEL SEL DCH)))
                        ;; Restore the highlighting for selected lines that are above or below the bitmap. The lines within the bitmap
                        ;; retained their proper highlighting. Above is first.
                        (for L (L1 _ (\TEDIT.SEL.L1 SEL PANE TEXTOBJ))
                             (SEL1 _ (FGETSEL SEL CH#))
                             (SELN _ (FGETSEL SEL CHLAST))
                            backlines
                            (FGETLD BMTOPL PREVLINE) first (CL:UNLESS (AND L1 (NEQ L1 BMTOPL)
                                                                           (IGREATERP (FGETLD L1 YTOP)
```

```
                                                                    BMTOPY))
                                                        ;; Selection's L1 is below the bitmap's new top.
                                                        (RETURN))
                                    when (FLINESELECTEDP L SEL1 SELN) do (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ L1 L PANE
                                                                                SEL)
                                                           (RETURN)
                                    repeatuntil (EQ L L1))
                                 (for L (LN _ (\TEDIT.SEL.LN SEL PANE TEXTOBJ))
                                     (SEL1 _ (FGETSEL SEL CH#))
                                     (SELN _ (FGETSEL SEL CHLAST))
                                     inlines
                                     (FGETLD BMBOTL NEXTLINE) first (CL:UNLESS (AND LN (ILESSP (FGETLD LN YBOT)
                                                                                               (IPLUS BMBOTY DELTA)))
                                                        ;; Selection's LN is above the bitmap's new bottom
                                                        (RETURN))
                                    when (FLINESELECTEDP L SEL1 SELN) do (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ L LN PANE
                                                                                SEL)
                                                           (RETURN)
                                    repeatuntil (EQ L LN)))]
          else ;; No useful bitmap bits, just create/display lines below PREVLINE
               (\TEDIT.CLEARPANE.BELOW.LINE PREVLINE PANE TEXTOBJ)
               (SETQ LASTVISIBLE (\TEDIT.LINES.BELOW PREVLINE PANE TEXTOBJ))
               (\TEDIT.SUFFIXLINE.CREATE PANE TEXTOBJ LASTVISIBLE)
               (\TEDIT.FIXSEL NIL TEXTOBJ NIL PANE))
     (CL:WHEN SCROLLING
          ;; If scrolling up or down, we brute force wipe out whatever is above PREVLINE.  If not scrolling, those are the lines from the top to
          ;; lastvalid that are preserved.
          (BLTSHADE WHITESHADE PANE (PANELEFT PANE)
                 (FGETLD PREVLINE YBOT)
                 (PANEWIDTH PANE)
                 (PANEHEIGHT PANE)
                 'REPLACE))
     ;; Caller is responsible for turning the caret back on
     (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ PANE])

)

(DEFINEQ
```

(\TEDIT.ONSCREEN?

```
  [LAMBDA (PANE PROP)                                                ; Edited  1-Dec-2024 11:50 by rmk
                                                                     ; Edited 21-Oct-2024 00:33 by rmk
                                                                     ; Edited 13-Jun-2024 22:12 by rmk
                                                                     ; Edited 15-May-2024 09:27 by rmk
                                                                     ; Edited 25-Apr-2024 22:23 by rmk
                                                                     ; Edited 22-Apr-2024 21:56 by rmk
                                                                     ; Edited  9-Apr-2024 16:57 by rmk
                                                                     ; Edited 29-Nov-2023 23:39 by rmk
    ;; If the PROP of PANE is on screen, returns that property of its clipping region (equivalent to (fetch PROP of (DSPCLIPPINGREGION NIL PANE).
    ;;  But if that property is off screen, the value gives the position in clipping region coordinates that is still visible.  E.g. if the bottom is below the
    ;; screen by N points (PREG's bottom is -N), the return is the clipping regions bottom -N.
    (LET [VAL (PREG (PANEREGION PANE))
         (REG (fetch (WINDOW REG) of PANE))
         (BORDER (OR 0 (WINDOWPROP PANE 'BORDER]
      (SELECTQ PROP
         (BOTTOM (SETQ VAL (fetch (REGION BOTTOM) of REG))
                 (if (ILESSP VAL 0)
                     then (IPLUS BORDER (IDIFFERENCE (fetch (REGION BOTTOM) of PREG)
                                                     VAL))
                   else (fetch (REGION BOTTOM) of PREG)))
         (TOP (SETQ VAL (fetch (REGION TOP) of REG))
              (if (IGREATERP VAL SCREENHEIGHT)
                  then (IDIFFERENCE (IDIFFERENCE (fetch (REGION TOP) of PREG)
                                                 (IDIFFERENCE VAL SCREENHEIGHT))
                                    (WINDOWPROP PANE 'BORDER))
                else (fetch (REGION TOP) of PREG)))
         (LEFT (SETQ VAL (fetch (REGION LEFT) of REG))
               (if (ILESSP VAL 0)
                   then ;; I don't understand why the border is subtracted instead of added.  But this makes \TEDIT.SCROLLUP and
                        ;; \TEDIT.SCROLLDOWN
                        (IDIFFERENCE (IDIFFERENCE (fetch (REGION LEFT) of PREG)
                                                  VAL)
                                     (WINDOWPROP PANE 'BORDER))
                 else (fetch (REGION LEFT) of PREG)))
         (RIGHT (SETQ VAL (fetch (REGION RIGHT) of REG))
                (if (IGREATERP VAL SCREENWIDTH)
                    then (IDIFFERENCE (IDIFFERENCE (fetch (REGION RIGHT) of PREG)
                                                   (IDIFFERENCE VAL SCREENWIDTH))
                                      (WINDOWPROP PANE 'BORDER))
                  else (fetch (REGION RIGHT) of PREG)))
```

```
            (HEIGHT                                                    ; Height of the onscreen subregion
                    (IDIFFERENCE (\TEDIT.ONSCREEN? PANE 'PTOP)
                           (\TEDIT.ONSCREEN? PANE 'BOTTOM)))
            (WIDTH (IDIFFERENCE (\TEDIT.ONSCREEN? PANE 'PRIGHT)
                          (\TEDIT.ONSCREEN? PANE 'LEFT)))
            (PTOP (SETQ VAL (fetch (REGION PTOP) of REG))
                  (if (IGREATERP VAL SCREENHEIGHT)
                      then (IDIFFERENCE (IDIFFERENCE (fetch (REGION PTOP) of PREG)
                                                     (IDIFFERENCE VAL SCREENHEIGHT))
                                        (WINDOWPROP PANE 'BORDER))
                    else (fetch (REGION PTOP) of PREG)))
            (PRIGHT (SETQ VAL (fetch (REGION PRIGHT) of REG))
                    (if (IGREATERP VAL SCREENWIDTH)
                        then (IDIFFERENCE (IDIFFERENCE (fetch (REGION PRIGHT) of PREG)
                                                       (IDIFFERENCE VAL SCREENWIDTH))
                                          (WINDOWPROP PANE 'BORDER))
                      else (fetch (REGION PRIGHT) of PREG)))
            (\TEDIT.THELP])
```

(\TEDIT.ONSCREEN.REGION

```
  [LAMBDA (WINDOW)                                                     ; Edited  1-Dec-2024 11:50 by rmk
                                                                       ; Edited 21-Oct-2024 00:27 by rmk
                                                                       ; Edited 13-Jun-2024 22:02 by rmk
                                                                       ; Edited 16-May-2024 17:40 by rmk
                                                                       ; Edited  6-May-2024 11:31 by rmk
    ;; This returns the intersection of WINDOW's clipping region and the screen region, in window coordinates. This is the part of the clipping region
    ;; that is visible on the screen, and presumably has a valid bitmap.

    ;; If WINDOW is entirely onscreen, the result will be a copy of the clipping region.Otherwise, this returns the subregion of the clipping region that is
    ;; actually visible.

    ;; For example, if the bottom of the PANE is below the screen by N points (PREG's bottom is -N), then VBOTTOM is clipping region's bottom + |N|.

    (\TEDIT.THELP "FIX WIDTH AND HEIGHT")
    (NOTUSED)
    (DECLARE (USEDFREE SCREEN))
    (if (WINDOWPROP WINDOW 'OFFSCREEN)
        then (LET ((CREG (DSPCLIPPINGREGION NIL WINDOW))
                   [BORDER (OR 0 (WINDOWPROP WINDOW 'BORDER]
                   (REG (fetch (WINDOW REG) of WINDOW))
                   VAL VLEFT VBOTTOM VWIDTH VHEIGHT)
                  (SETQ VAL (fetch (REGION BOTTOM) of REG))
                  (SETQ VBOTTOM (if (ILESSP VAL 0)
                                    then (IPLUS VAL BORDER (IDIFFERENCE (fetch (REGION BOTTOM) of CREG)))
                                  else (fetch (REGION BOTTOM) of CREG)))
                  (SETQ VAL (fetch (REGION LEFT) of REG))
                  (SETQ VLEFT (if (ILESSP VAL 0)
                                  then
                                       ;; I don't understand why the border is subtracted instead of added.  But this makes
                                       ;; \TEDIT.SCROLLUP and \TEDIT.SCROLLDOWN
                                       (IDIFFERENCE (IDIFFERENCE (fetch LEFT of CREG)
                                                                 VAL)
                                                    BORDER)
                                else (fetch (REGION LEFT) of CREG)))
                  (SETQ VAL (fetch (REGION WIDTH) of CREG))
                  (SETQ VWIDTH (if (IGREATERP (IDIFFERENCE (fetch (REGION RIGHT) of REG)
                                                           BORDER)
                                              (fetch (SCREEN SCWIDTH) of SCREEN))
                                   then (IDIFFERENCE (\TEDIT.ONSCREEN? WINDOW 'PRIGHT)
                                                     VLEFT)
                                 else VAL))
                  (SETQ VAL (fetch (REGION HEIGHT) of CREG))
                  (SETQ VHEIGHT (if (IGREATERP VAL SCREENHEIGHT)
                                    then (IDIFFERENCE (\TEDIT.ONSCREEN? WINDOW 'PTOP)
                                                      VBOTTOM)
                                  else VAL))
                  (create REGION
                          LEFT _ VLEFT
                          BOTTOM _ VBOTTOM
                          WIDTH _ VWIDTH
                          HEIGHT _ VHEIGHT))
      else (DSPCLIPPINGREGION NIL WINDOW])
```

(\TEDIT.AFTERMOVEFN

```
  [LAMBDA (PANE)                                                       ; Edited  6-Jul-2024 16:58 by rmk
                                                                       ; Edited 20-Jan-2024 23:22 by rmk
                                                                       ; Edited 21-Dec-2023 17:18 by rmk
                                                                       ; Edited 20-Dec-2023 00:33 by rmk
    ;; If PANE was partly off screen before this move, then repaint it.  If it is still off screen after this move, record that for the next move.

    (CL:WHEN (WINDOWPROP PANE 'OFFSCREEN)
           (\TEDIT.FILL.PANES PANE))                                   ; This repaints all PANE's sister, maybe not needed?
    (WINDOWPROP PANE 'OFFSCREEN (OFFSCREENP PANE])
```

(OFFSCREENP

```
  [LAMBDA (WINDOW)                                                    ; Edited 13-Jun-2024 21:59 by rmk
                                                                      ; Edited 19-Mar-2024 23:30 by rmk
                                                                      ; Edited 20-Jan-2024 23:23 by rmk
                                                                      ; Edited 21-Dec-2023 17:17 by rmk
                                                                      ; Edited 17-Dec-2023 17:27 by rmk
    ;; Returns a list indicating which boundaries of the window is offscreen.  Also includes VERTICAL or HORIZONTAL, if one of those respective
    ;; dimensions is off.
    (LET ((REGION (WINDOWREGION WINDOW))
          (SCREEN (fetch (WINDOW SCREEN) of WINDOW))
          RESULT)
         (CL:WHEN (ILESSP (fetch (REGION LEFT) of REGION)
                          0)
             (PUSH RESULT 'LEFT))
         (CL:WHEN (IGREATERP (fetch (REGION RIGHT) of REGION)
                             (fetch (SCREEN SCWIDTH) of SCREEN))
             (PUSH RESULT 'RIGHT))
         (CL:WHEN (OR (MEMB 'LEFT RESULT)
                      (MEMB 'RIGHT RESULT))
             (PUSH RESULT 'HORIZONTAL))
         (CL:WHEN (IGREATERP (fetch (REGION TOP) of REGION)
                             (fetch (SCREEN SCHEIGHT) of SCREEN))
             (PUSH RESULT 'TOP))
         (CL:WHEN (ILESSP (fetch (REGION BOTTOM) of REGION)
                          0)
             (PUSH RESULT 'BOTTOM))
         (CL:WHEN (OR (MEMB 'TOP RESULT)
                      (MEMB 'BOTTOM RESULT))
             (PUSH RESULT 'VERTICAL))
        RESULT])

)

;; Process-world interfaces

(DEFINEQ

(\TEDIT.PROCIDLEFN
  [LAMBDA (WINDOW)                                                    ; Edited 15-Mar-2024 18:34 by rmk
                                                                      ; Edited 25-Sep-2023 10:30 by rmk
                                                                      ; Edited 19-Sep-2023 23:25 by rmk
                                                                      ; Edited 30-May-91 23:35 by jds
    ;; TEDIT's PROC.IDLEFN for regaining control.  If the shift key is down, we're not trying to restart this window, just to copy from it.
    (GETMOUSESTATE)
    (COND
       ([AND (INSIDE? (DSPCLIPPINGREGION NIL WINDOW)
                      (LASTMOUSEX WINDOW)
                      (LASTMOUSEY WINDOW))
             [NOT (OR (SHIFTDOWNP 'SHIFT)
                      (SHIFTDOWNP 'META)
                      (KEYDOWNP 'MOVE)
                      (KEYDOWNP 'COPY]
             (PROCESSP (WINDOWPROP WINDOW 'PROCESS]
        (TTY.PROCESS (WINDOWPROP WINDOW 'PROCESS))                    ; No SHIFT key down;  let's regain control.
        (CL:WHEN (GETTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of WINDOW)
                          MENUFLG)                                    ; This is a MENU -- always select.
            (\TEDIT.MENU.BUTTONEVENTFN WINDOW)))
       (T                                                             ; Otherwise, let him select.
          (\TEDIT.BUTTONEVENTFN WINDOW])

(\TEDIT.PROCENTRYFN
  [LAMBDA (NEWPROCESS OLDPROCESS)                                     (* jds "15-Feb-84 16:59")
                                                                      (* TEDIT's PROCESS.ENTRYFN, which disarms any
                                                                      dangerous interrupts within the editing world)
    (\TEDIT.INTERRUPT.SETUP NEWPROCESS])

(\TEDIT.PROCEXITFN
  [LAMBDA (THISP NEWP)                                                ; Edited 27-Mar-2024 15:23 by rmk
                                                                      (* jds " 5-Apr-84 10:40")
    ;; Re-arm any interrupts that TEdit turned off, so the poor user has them available in other parts of the system.
          (* Re-arm any interrupts that TEdit turned off, so the poor user has them available in other parts of the system.)

    (AND (fetch (TEXTWINDOW WTEXTSTREAM) of (PROCESSPROP THISP 'WINDOW))
         (\TEDIT.INTERRUPT.SETUP THISP T])

)

(RPAQ? \CARETRATE 333)

;; Caret handler;  stolen from CHAT.

(DEFINEQ
```

(\TEDIT.DOWNCARET

```
[LAMBDA (CARET X Y)                                        ; Edited 26-Oct-2023 08:51 by rmk
                                                           ; Edited 13-Nov-87 08:25 by jds

    ;; Put the caret down -- i.e., MAKE IT VISIBLE -- as fast as possible

    (\DTEST CARET 'TEDITCARET)
    (LET ((DS (ffetch (TEDITCARET TCCARETDS) of CARET)))
         (freplace (TEDITCARET TCCARETX) of CARET with X)
         (DSPXPOSITION X DS)
         (freplace (TEDITCARET TCCARETY) of CARET with Y)
         (DSPYPOSITION Y DS)
         (freplace (TEDITCARET TCFORCEUP) of CARET with NIL)
         (\CARET.FLASH? DS (ffetch (TEDITCARET TCCARET) of CARET)
                10 NIL X Y])
```

(\TEDIT.FLASHCARET

```
[LAMBDA (TEXTOBJ)                                          ; Edited 28-Jun-2024 22:59 by rmk
                                                           ; Edited 25-May-2024 13:59 by rmk
                                                           ; Edited 19-Dec-2023 11:31 by rmk
                                                           ; Edited 12-Oct-2023 23:31 by rmk
                                                           ; Edited 16-Sep-2023 22:51 by rmk
                                                           (* jds "16-Jul-85 12:35")

    ;; Unless the caret is constrained to be INVISIBLE/UP, give it a chance to flash in every pane.

    (CL:UNLESS (AND NIL (FGETTOBJ TEXTOBJ TXTREADONLY))
        [bind (FIRSTTIME _ T) for PANE CARET inpanes TEXTOBJ eachtime (SETQ CARET (GETPANEPROP (PANEPROPS PANE)
                                                                                    PCARET))
           unless (fetch (TEDITCARET TCFORCEUP) of CARET) do       ; The caret need not stay invisible.
                                                          (if FIRSTTIME
                                                              then (SETQ FIRSTTIME NIL)
                                                                   (\CARET.FLASH? (fetch (TEDITCARET TCCARETDS)
                                                                                     of CARET)
                                                                          (fetch (TEDITCARET TCCARET)
                                                                             of CARET)
                                                                          NIL NIL (fetch (TEDITCARET TCCARETX)
                                                                                     of CARET)
                                                                          (fetch (TEDITCARET TCCARETY)
                                                                             of CARET))
                                                            else (\CARET.FLASH.AGAIN (fetch (TEDITCARET TCCARET)
                                                                                        of CARET)
                                                                       (fetch (TEDITCARET TCCARETDS)
                                                                          of CARET)
                                                                       (fetch (TEDITCARET TCCARETX) of CARET)
                                                                       (fetch (TEDITCARET TCCARETY) of CARET])])
```

(\TEDIT.UPCARET

```
[LAMBDA (CARET X Y)                                        ; Edited 26-Oct-2023 08:49 by rmk
                                                           ; Edited 12-Oct-2023 00:06 by rmk
                                                           ; Edited 13-Nov-87 08:27 by jds

    ;; Take the caret up -- i.e., MAKE IT INVISIBLE -- and keep it up.  Tedit and the system seem to have opposite notions of up and down.  System
    ;; thinks that caret is "down" when it is off the screen, Tedit thinks it is up.

    (\CARET.DOWN (fetch (TEDITCARET TCCARETDS) of CARET))

    ;; The TCFORCEUP field is set so that the caret will stay off-screen:

    (replace (TEDITCARET TCFORCEUP) of CARET with T)
    (CL:WHEN X
        (freplace (TEDITCARET TCCARETY) of CARET with X)
        (DSPXPOSITION X (ffetch (TEDITCARET TCCARETDS) of CARET))
        (freplace (TEDITCARET TCCARETY) of CARET with Y)
        (DSPYPOSITION Y (ffetch (TEDITCARET TCCARETDS) of CARET)))])
```

(TEDIT.NORMALIZECARET

```
[LAMBDA (TSTREAM SEL EVEN.IF.VISIBLE)                      ; Edited  9-Nov-2024 14:40 by rmk
                                                           ; Edited 29-Oct-2024 11:45 by rmk
                                                           ; Edited 21-Oct-2024 00:33 by rmk
                                                           ; Edited 21-Jun-2024 23:58 by rmk
                                                           ; Edited 19-Jun-2024 14:37 by rmk
                                                           ; Edited 16-Jun-2024 10:08 by rmk
                                                           ; Edited 16-May-2024 23:04 by rmk
                                                           ; Edited  7-May-2024 23:01 by rmk
                                                           ; Edited  1-May-2024 22:59 by rmk
                                                           ; Edited 29-Apr-2024 15:12 by rmk
                                                           ; Edited  9-Apr-2024 19:31 by rmk
                                                           ; Edited 21-Mar-2024 21:27 by rmk
                                                           ; Edited 21-Feb-2024 20:43 by rmk
                                                           ; Edited  2-Jan-2024 11:09 by rmk
                                                           ; Edited 20-Nov-2023 14:22 by rmk
                                                           ; Edited  5-Oct-2023 22:38 by rmk
                                                           ; Edited 11-May-2023 00:39 by rmk
                                                           ; Edited 30-May-91 23:35 by jds

    ;; This ensures that the caret is visible in the pane where the selection SEL was made.  Other panes are left alone (caret may or may not be
    ;; visible), presumably because you don't want all the panes to jump to the caret when you are working in just one of them.
```

```
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    (LET* ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
           (SELPANE (FGETTOBJ TEXTOBJ SELPANE)))
          (CL:UNLESS SEL
              (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
          (\DTEST SEL 'SELECTION)
          (CL:WHEN (AND SELPANE (FGETSEL SEL SET)
                        (IGREATERP (FGETTOBJ TEXTOBJ TEXTLEN)
                                   0))
              (CL:WHEN (OR EVEN.IF.VISIBLE (NOT (\TEDIT.VISIBLECARETP SELPANE TEXTOBJ)))
                  ;; Move the caret character to top of SELPANE.
                  (\TEDIT.SCROLLCH.TOP TSTREAM SELPANE (SELECTQ (FGETSEL SEL POINT)
                                                               (LEFT (FGETSEL SEL CH#))
                                                               (RIGHT (SUB1 (FGETSEL SEL CHLIM)))
                                                               (\TEDIT.THELP)))
                  TSTREAM))])
```

(\TEDIT.SETCARET

```
  [LAMBDA (SEL PANE TEXTOBJ DISPOSITION)                    ; Edited  1-Dec-2024 11:51 by rmk
                                                            ; Edited 22-Nov-2024 11:39 by rmk
                                                            ; Edited 20-Nov-2024 12:37 by rmk
                                                            ; Edited 17-Nov-2024 19:01 by rmk
                                                            ; Edited 21-Oct-2024 00:33 by rmk
                                                            ; Edited  9-Sep-2024 13:49 by rmk
                                                            ; Edited 28-Jun-2024 22:36 by rmk
                                                            ; Edited 25-May-2024 13:57 by rmk
                                                            ; Edited 15-Dec-2023 23:37 by rmk
                                                            ; Edited 17-Nov-2023 23:55 by rmk
                                                            ; Edited 16-Sep-2023 23:09 by rmk
                                                            ; Edited 20-Apr-2023 09:26 by rmk
                                                            ; Edited 30-May-91 23:35 by jds
    ;; Sets SEL's caret in PANE.  Caret will be located relative to L1 or LN in SEL, depending on POINT.

    ;; DISPOSITION:  NIL/OFF: suspend it temporarily but retain position, DISABLE:  move it out of the pane until reenabled, otherwise: flash at
    ;; POINT.

    (LET ((CARET (PANECARET PANE)))
         (CL:WHEN (FGETSEL SEL HASCARET)
             [SELECTQ DISPOSITION
                  ((NIL OFF DISABLE)
                       (\TEDIT.UPCARET CARET -10 -10))
                  (LET (LINE X Y)                           ; Is the caret on a visible line?
                       (for P inpanes (PROGN TEXTOBJ) as L1 in (FGETSEL SEL L1) as LN
                          in (FGETSEL SEL LN) when (AND L1 (EQ P PANE))
                          do ;; If L1 then at least one line of the selection is visible in PANE, and there is also a last visible line LN. But the
                             ;; caret shows only before the first character of the selection (X0) or the last character of the selection (XLIM)--we
                             ;; have to test to make sure we aren't look at intermediate lines.
                             (SELECTQ (FGETSEL SEL POINT)
                                  (LEFT (CL:WHEN (SETQ LINE (WITHINLINEP (FGETSEL SEL CH#)
                                                                         L1))
                                                 (SETQ X (FGETSEL SEL X0))))
                                  (RIGHT (CL:WHEN (SETQ LINE (WITHINLINEP (SUB1 (FGETSEL SEL CHLIM))
                                                                          LN))
                                                  (if (AND (FGETLD LINE FORCED-END)
                                                           (IEQP (FGETSEL SEL CHLIM)
                                                                 (FGETLD LINE LCHARLIM))
                                                           (FGETLD LINE NEXTLINE))
                                                      then
                                                      ;; Selection ends at an end-forcing character: caret flashes at the beginning of the next line, if there is
                                                      ;; one.  If not,we must be at the bottom of the pane or the end of the document.  We can create the
                                                      ;; line (maybe a dummy), but we also then have to scroll up so that it is visible. On input,
                                                      ;; TEDIT.NORMALIZECARET would do that.  But for selection, that would have to be in the
                                                      ;; DO.SELOPERATION code
                                                      (SETQ LINE (FGETLD LINE NEXTLINE))
                                                      (SETQ X (FGETLD LINE LX1))
                                                    else (SETQ X (FGETSEL SEL XLIM)))))
                                  (\TEDIT.THELP))
                             (RETURN))
                       (if LINE
                           then (SETQ Y (FGETLD LINE YBASE))
                                (if (AND (ILESSP Y (fetch (REGION PTOP) of (PANEREGION PANE)))
                                         (IGEQ (FGETLD LINE YBOT)
                                               (PANEBOTTOM PANE)))
                                    then                    ; Move to right position even if not flashing
                                         (CL:IF (AND NIL (FGETTOBJ TEXTOBJ TXTREADONLY))
                                             (\TEDIT.UPCARET CARET X Y)
                                             (\TEDIT.DOWNCARET CARET X Y))
                                  else ;; Caret line is not in PANE, make it go away
                                       (\TEDIT.UPCARET CARET -10 -10))
                         else ;; Disable if the intended line isn't visible. Maybe leave it at current position?
                              (\TEDIT.UPCARET CARET -10 -10])
```

```
         CARET])
```

(\TEDIT.CARET

```
  [LAMBDA (TEXTOBJ CARET)                                    ; Edited 30-Nov-2024 11:45 by rmk
                                                             ; Edited 27-Nov-2024 19:43 by rmk
                                                             ; Edited 28-Jun-2024 23:01 by rmk
                                                             (* jds "12-Jul-85 11:18")
    ;; Change the shape of the caret in each pane
    (for P inpanes TEXTOBJ do (replace TCCARET of (PANECARET P) with (\CARET.CREATE CARET)))
    CARET])

)

;; Menu interfacing

(DEFINEQ
```

(TEDIT.ADD.MENUITEM

```
  [LAMBDA (MENU ITEM)                                        (* jds " 9-AUG-83 09:55")
                                                             (* Adds ITEM to the MENU, and updates all the stuff.)
    (PROG (OLDITM)
          (COND
             ((MEMBER ITEM (fetch ITEMS of MENU))            (* Do nothing--it's already in the menu)
              )
             ([AND (LISTP ITEM)
                   (SETQ OLDITM (SASSOC (CAR ITEM)
                                        (fetch ITEMS of MENU]
                                                             (* The menu item exists. Make sure the thing behind it is right.)
              (RPLACD OLDITM (CDR ITEM)))
             (T                                              (* It isn't in the menu, so go ahead and add it.)
                (replace ITEMS of MENU with (NCONC1 (fetch ITEMS of MENU)
                                                    ITEM))
                (COND
                   ((EQ (fetch MENUCOLUMNS of MENU)
                        1)                                   (* If there is only one column, force a re-figuring of the number
                                                             of rows)
                    (replace MENUROWS of MENU with NIL))
                   ((EQ (fetch MENUROWS of MENU)
                        1)                                   (* There's only one row, so recompute %# of columns.)
                    (replace MENUCOLUMNS of MENU with NIL)))
                (replace ITEMWIDTH of MENU with 10000)
                (replace ITEMHEIGHT of MENU with 10000)
                (replace IMAGE of MENU with NIL)             (* Force it to create a new menu image.)
                (UPDATE/MENU/IMAGE MENU])
```

(TEDIT.DEFAULT.MENUFN

```
  [LAMBDA (PANE)                                             ; Edited 12-Feb-2025 16:26 by rmk
                                                             ; Edited  9-Feb-2025 21:28 by rmk
                                                             ; Edited  7-Jan-2025 23:46 by rmk
                                                             ; Edited 27-Jul-2024 20:24 by rmk
                                                             ; Edited 30-Jun-2024 12:38 by rmk
                                                             ; Edited 25-Jun-2024 11:59 by rmk
                                                             ; Edited 18-May-2024 16:50 by rmk
                                                             ; Edited 12-May-2024 21:38 by rmk
                                                             ; Edited 20-Mar-2024 11:02 by rmk
                                                             ; Edited 24-Apr-2024 09:47 by rmk
                                                             ; Edited 15-Mar-2024 18:35 by rmk
                                                             ; Edited  9-Mar-2024 11:35 by rmk
                                                             ; Edited 29-Feb-2024 17:02 by rmk
                                                             ; Edited 27-Feb-2024 07:55 by rmk
                                                             ; Edited 22-Sep-2023 20:14 by rmk
                                                             ; Edited 19-Sep-2023 11:55 by rmk
                                                             ; Edited 16-Sep-2023 22:16 by rmk
                                                             ; Edited  6-May-2023 17:28 by rmk
                                                             ; Edited 30-May-91 23:35 by jds
    ;; Default MENU Fn for editor windows--displays a menu of items & acts on the commands received.
    (PROG* ((TSTREAM (TEXTSTREAM PANE))
            (TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))
            (WMENU (WINDOWPROP PANE 'TEDIT.MENU))
            THISMENU ITEM)
           (CL:WHEN (FGETTOBJ TEXTOBJ EDITOPACTIVE)
                ;; We're busy doing something, tell him to wait.  Unfortunately, this string will overwrite whatever may be in the Tedit promptwindow
                ;; (e.g. a GETINPUT calling TTYINPROMPTFORWORD for a meta-F command), obscuring what the user has already typed.
                ;; Maybe an interface that tests to see if the promptwindow is in use, and enlarges it with an extra line above the current type-in?
                (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (CL:IF (EQ T (FGETTOBJ TEXTOBJ EDITOPACTIVE))
                                                          "Edit"
                                                          (FGETTOBJ TEXTOBJ EDITOPACTIVE))
                                                   " operation in progress; please wait")
                      T)
                (RETURN NIL))
           (SETQ THISMENU (if WMENU
                              elseif (SETQ WMENU (WINDOWPROP PANE 'TEDIT.MENU.COMMANDS))
                                then (PROG1 (SETQ WMENU (\TEDIT.CREATEMENU WMENU))
```

```
                                    (WINDOWPROP PANE 'TEDIT.MENU WMENU))
                           else TEDIT.DEFAULT.MENU))
          (SETQ ITEM (CAR (MENU THISMENU)))
          (ERSETQ (RESETLST
                      [SELECTQ ITEM
                           ((Put |Put Formatted Document|)
                               (TEDIT.PUT TEXTOBJ NIL NIL (GETTEXTPROP TEXTOBJ 'CLEARPUT)))
                           (Plain-Text (TEDIT.PUT TEXTOBJ NIL NIL T))
                           ((Get |Get Formatted Document|)      ; Get a new file (overwriting the one being edited.)
                               (TEDIT.GET TEXTOBJ NIL (GETTEXTPROP TEXTOBJ 'CLEARGET)))
                           (Unformatted% Get
                               (TEDIT.GET TEXTOBJ NIL T))
                           (Include                             ; Insert a file where the caret is
                                   (TEDIT.INCLUDE TEXTOBJ))
                           (Quit                                ; OK to stop this session?
                                 (\TEDIT.FINISHEDIT? TEXTOBJ))
                           (Substitute                          ; Search-and-replace
                                      (RESETLST
                                          (RESETSAVE (CURSOR WAITINGCURSOR))
                                          (TEDIT.SUBSTITUTE TEXTOBJ)))
                           (Find                                ; Case sensitive search, with * and # wildcards
                                 (\TEDIT.KEY.FIND TSTREAM TEXTOBJ))
                           (Looks                               ; He wants to set the font for the current selection
                                  (\TEDIT.LOOKS TEXTOBJ))
                           (Hardcopy                            ; Print this document
                                    (TEDIT.HARDCOPY TEXTOBJ))
                           (Expanded% Menu                      ; Open the expanded operations menu.
                               (\TEDIT.EXPANDEDMENU.START TEXTOBJ))
                           (Character% Looks                    ; Open the menu for setting character looks
                               (\TEDIT.CHARMENU.START TEXTOBJ))
                           (Paragraph% Formatting               ; Open the paragraph formatting menu
                               (\TEDIT.PARAMENU.START TEXTOBJ))
                           (Page% Layout                        ; Open the page-layout menu
                                        (\TEDIT.MENU.START (\TEDIT.PAGEMENU.CREATE)
                                                (\TEDIT.PRIMARYPANE TEXTOBJ)
                                                "Page Layout Menu" 150 'PAGE))
                           (CL:WHEN ITEM                        ; Apply a user-supplied function to the text stream
                               [RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ T)
                                      '(PROGN (\TEDIT.MARKINACTIVE OLDVALUE]
                               (APPLY* ITEM (fetch (TEXTWINDOW WTEXTSTREAM) of PANE)))])])
```

(TEDIT.REMOVE.MENUITEM

```
[LAMBDA (MENU ITEM)                                             (* gbn "26-Apr-84 04:06")
  (PROG (ITEMLIST)
        [COND
           ((OR (LITATOM ITEM)
                (STRINGP ITEM))
            (for X in (fetch ITEMS of MENU) do (COND
                                                  ((AND (LISTP X)
                                                        (EQUAL (CAR X)
                                                               ITEM))
                                                   (RETURN (SETQ ITEM X]
        (RETURN (COND
                   ((MEMBER ITEM (SETQ ITEMLIST (fetch ITEMS of MENU)))
                    (replace ITEMS of MENU with (REMOVE ITEM ITEMLIST))
                    (replace MENUCOLUMNS of MENU with NIL)
                    (replace MENUROWS of MENU with NIL)
                    (UPDATE/MENU/IMAGE MENU))
                   (T NIL])
```

(\TEDIT.CREATEMENU

```
[LAMBDA (ITEMS)                                                 ; Edited  3-Apr-2024 13:30 by rmk
                                                                ; Edited 16-Oct-87 14:21 by jds
```

;; Create a TEdit command menu, given a list of menu items.

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

(\TEDIT.MENU.WHENHELDFN

```
[LAMBDA (ITEM MENU BUTTON)                                      ; Edited  4-Oct-2022 09:17 by rmk
                                                                (* jds "10-Apr-84 15:14")
  (COND
     ((ATOM ITEM)
      (CLRPROMPT)
      (PROMPTPRINT (SELECTQ ITEM
                        (Put "Sends the document to a file")
                        (Get "Gets a new file as the document to edit.")
                        (Looks "Changes the font/size/etc. of characters")
                        (Find "Searches for a string")
                        (Quit "Ends the edit session")
```

```
                        (Hardcopy "Formats and sends the file to a printer.")
                        (Hardcopy% File
                            "Creates a hardcopy-format file of the document.")
                        """)))
       (T (DEFAULTMENUHELDFN ITEM])
```

(\TEDIT.MENU.WHENSELECTEDFN

```
  [LAMBDA (ITEM MENU BUTTON)                                    ; Edited 16-Oct-87 14:21 by jds
```

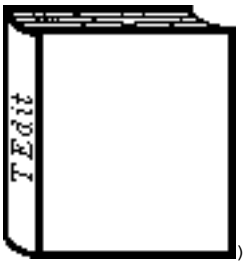
;; A Selection fn for preserving the button pressed, for special handling in PUT, e.g.

```
    (CONS (DEFAULTWHENSELECTEDFN ITEM MENU BUTTON)
          BUTTON])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.DEFAULT.MENU)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQ TEDIT.DEFAULT.MENU
      [\TEDIT.CREATEMENU '((Put 'Put NIL (SUBITEMS |Put Formatted Document| Plain-Text))
                          (Get 'Get NIL (SUBITEMS |Get Formatted Document| Unformatted% Get))
                          Include Find Looks Substitute Quit (Expanded% Menu 'Expanded% Menu NIL
                                                                (SUBITEMS Expanded% Menu Character% Looks
                                                                      Paragraph% Formatting Page% Layout])
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

[OR (SASSOC 'TEdit BackgroundMenuCommands)
    (NCONC1 BackgroundMenuCommands '(TEdit '(TEDIT)
                                        "Opens a TEdit window for use."]

(SETQ BackgroundMenu NIL)
)
```

;; titled icon info,

```
(FILESLOAD ICONW)
```

(RPAQQ **TEDITICON**



)

(RPAQQ **TEDITMASK**

)

```
(RPAQ? TEDIT.ICON.FONT (FONTCREATE 'HELVETICA 8 'BOLD))

(RPAQ? TEDIT.ICON.TITLE.REGION (CREATEREGION 16 4 64 77))

(RPAQ? TEDIT.TITLED.ICON.TEMPLATE (create TITLEDICON ICON _ TEDITICON MASK _ TEDITMASK TITLEREG _
                                          TEDIT.ICON.TITLE.REGION))
```

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX

I.S.OPR INDEX