File created: 27-Mar-2024 15:23:52 **{WMEDLEY}<library>tedit>TEDIT-WINDOW.;407**

edit by: rmk

changes to: (FNS \TEDIT.PROCEXITFN \TEDIT.BUTTONEVENTFN)

previous date: 21-Mar-2024 21:27:34 {WMEDLEY}<library>TEDIT>TEDIT-WINDOW.;405

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-WINDOWCOMS
       [(DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS TEDITCARET TEXTWINDOW PANE)
                                                (MACROS FGETPANE GETPANE SETPANE FSETPANE)
                                                (I.S.OPRS inpanes)))
        (INITRECORDS TEDITCARET PANE)
        (FILES ATTACHEDWINDOW)
        (FNS \TEDIT.CREATEW \TEDIT.WINDOW.SETUP \TEDIT.MINIMAL.WINDOW.SETUP \TEDIT.ADD.CARET \TEDIT.CLEARPANE)
        (FNS \TEDIT.CURSORMOVEDFN \TEDIT.CURSOROUTFN \TEDIT.ACTIVE.WINDOWP \TEDIT.EXPANDFN \TEDIT.MAINW
             \TEDIT.PRIMARYW \TEDIT.NEWREGIONFN \TEDIT.SET.WINDOW.EXTENT \TEDIT.SHRINK.ICONCREATE
             \TEDIT.SHRINKFN \TEDIT.PANEREGION)
        (FNS \TEDIT.BUTTONEVENTFN \TEDIT.DO.SELOPERATION \TEDIT.TTY.TEXTOBJP \TEDIT.BUTTONEVENTFN.SELOPERATION
             \TEDIT.BUTTONEVENTFN.INACTIVE \TEDIT.BUTTONEVENTFN.INTITLE \TEDIT.COPYINSERT)
        (P (MOVD? 'NILL '\TEDIT.COPYINSERT))
        (FNS \TEDIT.PANE.SPLIT \TEDIT.SPLITW \TEDIT.UNSPLITW)
        (P (MOVD? 'NILL 'GRAB-TYPED-REGION)
           (MOVD? 'NILL 'REGISTER-TYPED-REGION))
        (INITVARS (\TEDIT.OP.WIDTH 12)
               (\TEDIT.OP.BOTTOM 12)
               (\TEDIT.LINEREGION.WIDTH 8))
        (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.REPAINTFN \TEDIT.AFTERMOVEFN OFFSCREENP \TEDIT.RESHAPEFN
                   \TEDIT.PANEWITHINSCREEN?)
              (FNS \TEDIT.SCROLLFN \TEDIT.SCROLLFLOAT \TEDIT.SCROLLUP \TEDIT.SCROLL.SHOWSEL \TEDIT.SCROLLDOWN
                   \TEDIT.OFFSCREEN.SCROLL \TEDIT.WHERE.SEL \TEDIT.WHERE.SEL1)
              (FNS \TEDIT.ONSCREEN \TEDIT.ONSCREEN? \TEDIT.PANE.SCREENREGION))
        (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 ((NEXTPANE (GETWINDOWPROP DATUM 'TEDIT-NEXT-PANE-DOWN)
                              (PUTWINDOWPROP DATUM 'TEDIT-NEXT-PANE-DOWN NEWVALUE))
                       (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)))
                       (WLINES (GETWINDOWPROP DATUM 'LINES)
                              (PUTWINDOWPROP DATUM 'LINES NEWVALUE))
                       (CURSORREGION (GETWINDOWPROP DATUM 'TEDIT.CURSORREGION)
                              (PUTWINDOWPROP DATUM 'TEDIT.CURSORREGION NEWVALUE))
                       (PLINES (GETWINDOWPROP DATUM 'LINES)
                              (PUTWINDOWPROP DATUM 'LINES NEWVALUE))
                       (CLOSINGFILE (GETWINDOWPROP DATUM 'TEDIT-CLOSING-FILE)
                              (PUTWINDOWPROP DATUM 'TEDIT-CLOSING-FILE NIL))
                       (WITHINSCREEN (GETWINDOWPROP DATUM 'TEDIT-WITHIN-SCREEN)
                              (LET ((NV NEWVALUE))
                                   (PUTWINDOWPROP DATUM 'TEDIT-WITHIN-SCREEN NV)
                                   NV]

[DATATYPE PANE ((XPWINDOW FULLXPOINTER)
                PLINES PCARET HOLDDUMMYFIRSTLINE NEXTPANE (PREVPANE XPOINTER))
       (ACCESSFNS (PWINDOW (PROGN DATUM]
)

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

(/DECLAREDATATYPE 'PANE '(FULLXPOINTER POINTER POINTER POINTER POINTER XPOINTER)
       ;; ---field descriptor list elided by lister---
       '12)

(DECLARE%: EVAL@COMPILE

(PUTPROPS FGETPANE MACRO ((P FIELD)
                           (ffetch (PANE FIELD) of P)))

(PUTPROPS GETPANE MACRO ((P FIELD)
                          (fetch (PANE FIELD) of P)))

(PUTPROPS SETPANE MACRO ((P FIELD NEWVALUE)
                          (replace (PANE FIELD) of P with NEWVALUE)))

(PUTPROPS FSETPANE MACRO ((P FIELD NEWVALUE)
                           (freplace (PANE FIELD) of P with NEWVALUE)))
)

(DECLARE%: EVAL@COMPILE
[I.S.OPR 'inpanes NIL '(inside (fetch (TEXTOBJ \WINDOW) of BODY]
)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'TEDITCARET '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                                            POINTER)

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

       '22)

(/DECLAREDATATYPE 'PANE '(FULLXPOINTER POINTER POINTER POINTER POINTER XPOINTER)

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

       '12)

(FILESLOAD ATTACHEDWINDOW)

(DEFINEQ

(\TEDIT.CREATEW
  [LAMBDA (WINDOW TSTREAM PROPS)                                ; 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)
         (CL:WHEN (WINDOWP WINDOW)
             (CL:WHEN (SETQ WTEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of WINDOW))

                 ;; Reusing an existing Tedit window, undo its splits.

                 (for P in (REVERSE (CDR (FGETTOBJ WTEXTOBJ \WINDOW))) 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 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 REGION (GETREGION 32 (IPLUS PHEIGHT 32)
                                     REGIONTYPE))               ; 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 (WINDOWPROP WINDOW 'PROMPTWINDOW)
                     '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 FIRSTLINE)                    ; Edited 15-Mar-2024 13:36 by rmk
                                                                      ; Edited  9-Feb-2024 10:51 by rmk
                                                                      ; Edited 29-Jan-2024 17:10 by rmk
                                                                      ; Edited 11-Jan-2024 19:33 by rmk
                                                                      ; Edited  2-Jan-2024 19:15 by rmk
                                                                      ; Edited 12-Oct-2023 23:41 by rmk
                                                                      ; Edited 10-Oct-2023 00:30 by rmk
                                                                      ; Edited  4-Oct-2023 22:59 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)
           (HELP "PANE=AFTERPANE"))
    (\DTEST PANE 'WINDOW)
    (LET ((TEXTOBJ (TEXTOBJ TSTREAM))
          (MENUPROP (LISTGET PROPS 'MENU))
          SEL PLINE)                                                  ; The Command menu, or list of items for it
         (COND
            ((type? MENU MENUPROP)                                    ; A menu.  just use it.
             (WINDOWPROP PANE 'TEDIT.MENU MENUPROP))
            (MENUPROP                                                 ; 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.CLEARPANE PANE)
         (SETQ PLINE (\TEDIT.CREATEPLINE TEXTOBJ PANE FIRSTLINE))
         (\TEDIT.ADD.CARET TEXTOBJ PANE AFTERPANE)
         (SETQ SEL (TEXTSEL TEXTOBJ))
         (\TEDIT.SHOWSEL SEL NIL (AND AFTERPANE PANE))
         (\TEDIT.FIXSEL SEL TEXTOBJ NIL (AND AFTERPANE PANE))
         (FSETSEL SEL HASCARET (NOT (FGETTOBJ TEXTOBJ TXTREADONLY)))
         (\TEDIT.SHOWSEL SEL T (AND AFTERPANE PANE])
```

(\TEDIT.MINIMAL.WINDOW.SETUP

```
  [LAMBDA (WINDOW TSTREAM PROPS AFTERPANE)                            ; 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 20-Jan-2024 23:24 by rmk
                                                                      ; Edited  2-Jan-2024 17:27 by rmk
                                                                      ; Edited 21-Dec-2023 17:19 by rmk
                                                                      ; Edited 17-Dec-2023 17:14 by rmk
                                                                      ; Edited  9-Dec-2023 20:14 by rmk
                                                                      ; Edited  3-Dec-2023 20:25 by rmk
                                                                      ; Edited 20-Nov-2023 10:40 by rmk
                                                                      ; Edited  4-Oct-2023 09:48 by rmk
                                                                      ; Edited 30-Sep-2023 17:36 by rmk
                                                                      ; Edited 21-Sep-2023 14:10 by rmk
                                                                      ; Edited 18-Sep-2023 23:44 by rmk
                                                                      ; Edited 30-May-91 23:33 by jds

    ;; Do the minimum setup so that WINDOW becomes a pane of TSTREAM and TSTREAM and WINDOW 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 WINDOW 'WINDOW)
(LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
      DS PROP OLDPANES)                                       ; The displaystream for flashing the caret
     (FSETTOBJ TEXTOBJ PANES (CONS (create PANE
                                           XPWINDOW _ WINDOW)))
                                                              ; NOT IMPLEMENTED YET
     (FSETTOBJ TEXTOBJ SELPANE WINDOW)
     (WINDOWPROP WINDOW 'PROCESS NIL)                         ; For the moment, this pane has no process
     (replace (TEXTWINDOW WTEXTSTREAM) of WINDOW with TSTREAM) ; TSTREAM is accessible from WINDOW
     (replace (TEXTWINDOW CURSORREGION) of WINDOW with (CREATEREGION 0 0 0 0))
                                                              ; Used by CursorMovedFn
     (SETQ DS (WINDOWPROP WINDOW 'DSP))
     (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))                ; Remember its size, too.
     [COND
        ((SETQ PROP (LISTGET PROPS 'REGION))                  ; Use the callers subregion
         (FSETTOBJ TEXTOBJ WTOP (fetch PTOP of PROP))
         (FSETTOBJ TEXTOBJ WRIGHT (fetch RIGHT of PROP))
         (FSETTOBJ TEXTOBJ WBOTTOM (fetch BOTTOM of PROP))
         (FSETTOBJ TEXTOBJ WLEFT (fetch LEFT of PROP)))
        (T                                                    ; Otherwise, default to the whole window
           (FSETTOBJ TEXTOBJ WLEFT (fetch LEFT of (DSPCLIPPINGREGION NIL DS)))
           (FSETTOBJ TEXTOBJ WBOTTOM (fetch BOTTOM of (DSPCLIPPINGREGION NIL DS)))
           (FSETTOBJ TEXTOBJ WTOP (fetch HEIGHT of (DSPCLIPPINGREGION NIL DS)))
           (FSETTOBJ TEXTOBJ WRIGHT (fetch WIDTH of (DSPCLIPPINGREGION NIL DS]
     (WINDOWPROP WINDOW 'CURSORMOVEDFN (FUNCTION \TEDIT.CURSORMOVEDFN))
     (WINDOWPROP WINDOW 'CURSOROUTFN (FUNCTION \TEDIT.CURSOROUTFN))
     (WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION \TEDIT.BUTTONEVENTFN))
     (WINDOWPROP WINDOW 'RIGHTBUTTONFN (FUNCTION \TEDIT.BUTTONEVENTFN))
     (WINDOWPROP WINDOW 'HARDCOPYFN (FUNCTION TEDIT.HARDCOPYFN))
     (WINDOWPROP WINDOW 'HARDCOPYFILEFN (FUNCTION \TEDIT.HARDCOPYFILEFN))
     (WINDOWADDPROP WINDOW 'RESHAPEFN (FUNCTION \TEDIT.RESHAPEFN))
     (WINDOWADDPROP WINDOW 'NEWREGIONFN (FUNCTION \TEDIT.NEWREGIONFN))
     (CL:UNLESS (WINDOWPROP WINDOW 'SCROLLFN)
         (WINDOWPROP WINDOW 'SCROLLFN (FUNCTION \TEDIT.SCROLLFN)))
     (WINDOWPROP WINDOW 'REPAINTFN (FUNCTION \TEDIT.REPAINTFN))
     (WINDOWPROP WINDOW 'AFTERMOVEFN (FUNCTION \TEDIT.AFTERMOVEFN))
     (WINDOWADDPROP WINDOW 'CLOSEFN (FUNCTION TEDIT.DEACTIVATE.WINDOW)
            T)                                                ; Clean up when the pane is closed
     (WINDOWPROP WINDOW 'WINDOWENTRYFN (FUNCTION \TEDIT.PROCIDLEFN))
                                                              ; Grab the TTY when the mouse clicks in the pane
     (WINDOWPROP WINDOW 'OFFSCREEN (OFFSCREENP WINDOW))       ; In case it is created off-screen
     (CL:UNLESS (WINDOWPROP WINDOW 'ICONFN)
         (WINDOWPROP WINDOW 'ICONFN (FUNCTION \TEDIT.SHRINK.ICONCREATE)))
                                                              ; Set up to create a shrink icon if nobody else has.
     (WINDOWADDPROP WINDOW 'SHRINKFN (FUNCTION \TEDIT.SHRINKFN))
                                                              ; Always give up control of the keyboard on shrinking.
     (WINDOWADDPROP WINDOW 'EXPANDFN (FUNCTION \TEDIT.EXPANDFN))
     (CL:UNLESS (WINDOWPROP WINDOW 'TEDIT.TITLEMENUFN)        ; The default menu fn
         (WINDOWPROP WINDOW 'TEDIT.TITLEMENUFN (OR (LISTGET PROPS 'TITLEMENUFN)
                                                   (FUNCTION TEDIT.DEFAULT.MENUFN))))
     (SETQ OLDPANES (FGETTOBJ TEXTOBJ \WINDOW))
     (CL:UNLESS (EQMEMB WINDOW OLDPANES)                      ; Don't
         (FSETTOBJ TEXTOBJ \WINDOW (if AFTERPANE
                                       then                   ; Put it after AFTERPANE
                                            [RPLACD (FMEMB AFTERPANE OLDPANES)
                                                   (CONS WINDOW (CDR (FMEMB AFTERPANE OLDPANES]
                                           OLDPANES
                                     else                     ; Otherwise, just add it at the end of the list
                                          (NCONC1 OLDPANES WINDOW))))
     WINDOW])
```

(\TEDIT.ADD.CARET

```
  [LAMBDA (TEXTOBJ PANE AFTERPANE)                            ; Edited 10-Mar-2024 15:01 by rmk
                                                              ; Edited  2-Jan-2024 19:21 by rmk
                                                              ; Edited  9-Oct-2023 22:40 by rmk
                                                              ; Edited  4-Oct-2023 23:38 by rmk
                                                              ; Edited 30-Sep-2023 23:57 by rmk
```

;; Adds a caret to TEXTOBJ that correspond to a new pane, maybe the very first one. TCFORCEUP is T to prevent the caret from flashing before
;; its position is known.

;; The OR handles the case where the CARET list has not yet been set up for the first pane.

```
    (if AFTERPANE
        then (for P inpanes (PROGN TEXTOBJ) as CTAIL on (FGETTOBJ TEXTOBJ CARET) when (EQ P AFTERPANE)
                do [PUSH (CDR CTAIL)
                         (create TEDITCARET
                                TCFORCEUP _ T
                                TCCARETDS _ (WINDOWPROP PANE 'DSP]
                   (RETURN T))
      else (FSETTOBJ TEXTOBJ CARET (CONS (create TEDITCARET
                                              TCFORCEUP _ T
                                              TCCARETDS _ (WINDOWPROP PANE 'DSP])
```

```
(\TEDIT.CLEARPANE
  [LAMBDA (PANE PBOTTOM)                                          ; Edited  2-Jan-2024 11:13 by rmk

    ;; Clears PANE's clipping region.  PBOTTOM is usually NIL, but can focus clearing on a subregion.

    (LET ((PREG (DSPCLIPPINGREGION NIL PANE)))
         (CL:UNLESS PBOTTOM
             (SETQ PBOTTOM (fetch (REGION BOTTOM) of PREG)))
         (BLTSHADE WHITESHADE PANE 0 PBOTTOM (fetch (REGION WIDTH) of PREG)
                (IDIFFERENCE (fetch (REGION PTOP) of PREG)
                       PBOTTOM)
                'REPLACE])
)

(DEFINEQ

(\TEDIT.CURSORMOVEDFN
  [LAMBDA (PANE)                                                  ; 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? (DSPCLIPPINGREGION NIL 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 (FGETTOBJ TEXTOBJ WBOTTOM)
                                      \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 LEFT of CURSORREG with LEFT)
                       (replace WIDTH of CURSORREG with \TEDIT.OP.WIDTH)
                else 
                     ;; Not in the split region.  Are we in the line-select region on the left?

                     (SETQ LINE (find L inlines (fetch (TEXTWINDOW PLINES) of PANE)
                                   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 LEFT of CURSORREG with 0)
                              (replace 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 LEFT of CURSORREG with LEFT)
                            (replace 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 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

    ;; The Tedit stream TSTREAM may have panes that are attached to other windows.  The typical case is a Tedit menu stream attached to a primary
    ;; editing window (\TEDIT.PRIMARYW), although in that case we wouldn't expect TSTREAM to have multiple panes.   This returns the main
    ;; window of that attachment (which may not be that editor's primary pane, if menus can be attached to window-splits (this should not be allowed).

    ;; The MEMB test deals with the fact that the panes of a split text stream are attached to each other--the MAINWINDOW of a later pane is the pane
    ;; before it--the primary (first) pane doesn't itself attach to a pane (although perhaps it could be attached to something else).

    (LET ((PANES (FGETTOBJ (TEXTOBJ TSTREAM)
                        \WINDOW)))
         (for PANE M inside PANES do (SETQ M (WINDOWPROP PANE 'MAINWINDOW))
                                     (if M
                                         then

                                       ;; This is attached to something.  If what it is attached to is one of TSTREAM's panes (e.g. the
                                       ;; preceding split), we keep going.  Presumably we eventually arrive at TSTREAM's main window
                                       ;; (e.g. the menu window), and its MAINWINDOW presumably is the primary window of an original
                                       ;; text stream.  I.e. we don't want to return one of our earlier panes.

                                              (CL:UNLESS (MEMB M PANES)
                                                     (RETURN M))
                                       else (RETURN PANE])
```

(\TEDIT.PRIMARYW

```
  [LAMBDA (TSTREAM)                                          ; Edited 19-Sep-2023 08:21 by rmk
                                                             ; Edited 30-May-91 23:33 by jds

    ;; This returns the first pane in the list of panes associated with TSTREAM.  Presumably this is the original pane, before any splitting.  Note that this
    ;; is different than \TEDIT.MAINW: that maps from attached windows (e.g. menus) back to a pane that they are attached to (presumably the
    ;; first/original pane).

    (CAR (MKLIST (GETTOBJ (TEXTOBJ TSTREAM)
                        \WINDOW])
```

(\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 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 (FIRSTLINE LASTLINE PHEIGHT PBOTTOM TOPCHAR BOTCHAR EXTHEIGHT EXTBOT YBOT
                        (TEXTLEN (FGETTOBJ TEXTOBJ TEXTLEN))
                        (PREG (DSPCLIPPINGREGION NIL PANE)))
                 (SETQ PHEIGHT (fetch HEIGHT of PREG))
                 (SETQ PBOTTOM (fetch BOTTOM of PREG))

                 ;; First visible line

                 (SETQ FIRSTLINE (find L inlines (fetch (TEXTWINDOW PLINES) of 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 LCHARLIM)))
                           (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 _ (fetch WIDTH of PREG)
                                                 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 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.

    (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])
)

(DEFINEQ

(\TEDIT.BUTTONEVENTFN
  [LAMBDA (PANE)                                                      ; Edited 27-Mar-2024 12:25 by rmk
                                                                  ; Edited 20-Mar-2024 11:01 by rmk
                                                                  ; Edited 16-Mar-2024 00:22 by rmk
                                                                  ; Edited  9-Mar-2024 11:59 by rmk
                                                                  ; Edited 24-Feb-2024 15:29 by rmk
                                                                  ; Edited 22-Feb-2024 14:57 by rmk
                                                                  ; Edited 19-Feb-2024 14:50 by rmk
                                                                  ; Edited 17-Feb-2024 15:40 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 button events for a TEdit pane.

    ;; RMK: 2021/9 TOTOPW was in (almost) all the conditional branches, I moved it up so that it always happens, even if the click is perhaps in a
    ;; menu.  There were cases where a second click in the window was needed to bring it above an overlapping window that it was under.  I think
    ;; perhaps it was because the mouse button may not have been seen as down on the first click, so it would return before it raised the window.  But
    ;; that was really bizarre--maybe the click was to see what was obscured by the overlapping window.

    (TOTOPW PANE)

    ;; Original code tested a global variable TEDIT.SELPENDING to prevent a selection happening while another one was pending, perhaps in a
    ;; different Tedit.   That variable held the textobj so that only the right command loop would act.  But now we set a variable directly in the
    ;; command-process associated with this text, so no other Tedits will see our selection.

    (CL:WHEN (MOUSESTATE (OR LEFT MIDDLE RIGHT))

        ;; If no button is down, we got control on button-up transition, so ignore it.

        (RESETLST
            [PROG ((TEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of PANE))
                   (DS (WINDOWPROP PANE 'DSP))
                   (X (LASTMOUSEX PANE))
                   (Y (LASTMOUSEY PANE))
                   SOURCESEL SELOPERATION SELFN)
                  (CL:UNLESS TEXTOBJ                                  ; Not a Tedit window
                      (RETURN))
                  (TEXTOBJ! TEXTOBJ)

             ;; Pick off and return from a bunch of peripheral situations, then fall through to the complexities of normal text selection.

                  (CL:WHEN (OR (\TEDIT.BUTTONEVENTFN.INTITLE Y PANE TEXTOBJ)
                               (\TEDIT.BUTTONEVENTFN.INACTIVE TEXTOBJ PANE)
                               (\TEDIT.PANE.SPLIT TEXTOBJ PANE))
                         (RETURN))

             ;;

             ;; The usual case -- he's really selecting something in this pane.  And there's nothing else going on now.

                  (\CARET.DOWN)                                       ; Make sure the caret isn't being displayed.
                                                                  ; Make the caret be the special, tall one so he can see it.
                  (RESETSAVE (for CARET in (GETTOBJ TEXTOBJ CARET) do (replace TCCARET of CARET with (\CARET.CREATE
                                                                                                          BXHICARET)))
                         (LIST '\TEDIT.CARET (GETTOBJ TEXTOBJ CARET)))
                  (SETQ SELFN (GETTEXTPROP TEXTOBJ 'SELFN))

             ;;

             ;; Polling loop, track the mouse until the buttons/keys come up.

                  (SETQ SELOPERATION (\TEDIT.BUTTONEVENTFN.SELOPERATION TEXTOBJ))
                  (bind (OSELOP _ SELOPERATION)
                        (OLDX _ MIN.SMALLP)
                        (OLDY _ MIN.SMALLP)
                        (PREG _ (DSPCLIPPINGREGION NIL PANE))
                       OSEL EXTENDFLG first (SETQ SOURCESEL (FGETTOBJ TEXTOBJ SCRATCHSEL))
                                                                  ; Get the storage and looks
                                           (SETQ OSEL (FGETTOBJ TEXTOBJ SCRATCHSEL2))
                                           (AND T (SETQ OSEL (create SELECTION
                                                                  SELTEXTOBJ _ TEXTOBJ)))
                                                                  ; TAKE THIS OUT WHEN SELTEXTOBJ GOES
                                           (\TEDIT.SET.SEL.LOOKS OSEL SELOPERATION)
                                           (SELECTQ SELOPERATION
                                               ((NORMAL DELETE)
                                                    (\TEDIT.COPYSEL (FGETTOBJ TEXTOBJ SEL)
                                                           SOURCESEL)
                                                    (\TEDIT.SHOWSEL (FGETTOBJ TEXTOBJ SEL)
                                                           NIL)
                                                    (\TEDIT.COPYSEL (FGETTOBJ TEXTOBJ SEL)
                                                           OSEL))
                                               (\TEDIT.SET.SEL.LOOKS SOURCESEL SELOPERATION))
                                           (FSETSEL OSEL CH# 0)
                    while [OR (SHIFTDOWNP 'SHIFT)
                               (SHIFTDOWNP 'CTRL)
                               (SHIFTDOWNP 'META)
                               (KEYDOWNP 'MOVE)
                               (KEYDOWNP 'COPY)
                               (NOT (ZEROP (LOGAND LASTMOUSEBUTTONS 7]
                    do                                               ; Poll the selection & display its current state
                        (if (ZEROP (LOGAND LASTMOUSEBUTTONS 7))
                            then                                      ; No mouse buttons are down;  don't try anything.
                                 (SETQ OLDX MIN.SMALLP)
                          else (SETQ SELOPERATION (\TEDIT.BUTTONEVENTFN.SELOPERATION TEXTOBJ)))
                        (if (AND (NOT (AND (IEQP OLDX (SETQ X (LASTMOUSEX DS)))
                                           (IEQP OLDY (SETQ Y (LASTMOUSEY DS)))
                                           (EQ OSELOP SELOPERATION)))
                                 (INSIDEP PREG X Y))
                            then ;; Only do selection if the mouse is inside the window proper and either the mouse has moved or the kind of
                                 ;; selection has changed

                                 ;; Must precede the scroll-region test, so that we don't try to scroll while the mouse is inside the main
                                 ;; window, even if the scroll bar overlaps the window (at left edge of screen, say)

                                 (SETQ OLDX X)
                                 (SETQ OLDY Y)
                                 (SETQ EXTENDFLG NIL)
                                 (if (\TEDIT.MOUSESTATE LEFT)
                                     then                                    ; Left selects char/point
                                          (SETQ SOURCESEL (\TEDIT.SELECT X Y TEXTOBJ (FGETTOBJ TEXTOBJ
                                                                                               MOUSEREGION)
                                                                 NIL SELOPERATION PANE))
                                   elseif (\TEDIT.MOUSESTATE MIDDLE)
                                     then                                    ; Middle selects word/line
                                          (SETQ SOURCESEL (\TEDIT.SELECT X Y TEXTOBJ (FGETTOBJ TEXTOBJ
                                                                                               MOUSEREGION)
                                                                 T SELOPERATION PANE))
                                   elseif (\TEDIT.MOUSESTATE RIGHT)
                                     then                                    ; RIght button extends last SOURCESEL
                                          (CL:UNLESS (EQ SELOPERATION OSELOP)

                                             ;; Things changed since the last selection.  Grab the prior selection info, so that the extension is taken
                                             ;; from the selection NOW being made, rather than the last existing old-type selection.

                                              (CL:WHEN OSEL (\TEDIT.COPYSEL OSEL SOURCESEL)))
                                          (SETQ SOURCESEL (\TEDIT.COPYSEL SOURCESEL))
                                          (CL:WHEN (AND TEDIT.EXTEND.PENDING.DELETE (EQ SELOPERATION 'NORMAL))

                                              ;; Simulate Laurel bluependingdelete: black, deletes on type-in

                                              (SETQ SELOPERATION 'PENDINGDEL)
                                              (\TEDIT.SET.SEL.LOOKS SOURCESEL 'DELETE))
                                          (SETQ SOURCESEL (\TEDIT.COPYSEL (\TEDIT.EXTEND.SEL X Y SOURCESEL
                                                                                 TEXTOBJ SELOPERATION PANE)))
                                          (SETQ EXTENDFLG T))
                                 (CL:WHEN [AND SELFN SOURCESEL (FGETSEL SOURCESEL SET)
                                               (EQ 'DON'T (APPLY* SELFN TEXTOBJ SOURCESEL SELOPERATION
                                                                  'TENTATIVE]

                                     ;; The selfn vetoed this selection, so mark it un-set and break out of the polling loop.

                                     (\TEDIT.SHOWSEL SOURCESEL NIL)
                                     (FSETSEL SOURCESEL SET NIL)
                                     (RETURN))
                                 (CL:WHEN OSEL
                                     (if (\TEDIT.SEL.CHANGED? SOURCESEL OSEL OSELOP SELOPERATION)
                                         then ;; Something interesting about the selection changed.  We have to re-display its image.

                                              (SETQ SOURCESEL (\TEDIT.REFRESH.SHOWSEL TEXTOBJ SOURCESEL OSEL
                                                                     OSELOP SELOPERATION EXTENDFLG))
                                              (SETQ OSELOP SELOPERATION)
                                       elseif (AND (FGETSEL OSEL SET)
                                                   (EQ (FGETSEL OSEL SELKIND)
                                                       'VOLATILE))
                                         then ;; THIS MAY BE OLD, FROM A GLOBAL SET ELSEWHERE ??  MENU?

                                           ;; There is an old selection around, but it is VOLATILE -- i.e., it shouldn't last longer than
                                           ;; something is pointing at it.  Turn it off.

                                              (\TEDIT.SHOWSEL OSEL NIL)
                                              (FSETSEL OSEL SET NIL)))
                                 (CL:WHEN SOURCESEL                          ; Maybe clicked in the boonies?
                                     (SETQ OSEL (\TEDIT.COPYSEL SOURCESEL OSEL)))
                          elseif (IN/SCROLL/BAR? PANE LASTMOUSEX LASTMOUSEY)
                            then                                             ; Mouse moved to scroll bar
                                 (SCROLL.HANDLER PANE))
                        (BLOCK)                                               ; Give other processes a chance
                        (GETMOUSESTATE)                                       ; And get the new mouse info
                        (\TEDIT.CURSORMOVEDFN PANE))

             ;; End Polling loop

             ;;

                  (CL:UNLESS (AND SOURCESEL (FGETSEL SOURCESEL SET))
                                                                              ; Bail if we didn't end up with an active selection
                      (RETURN))
                  (CL:UNLESS (INSIDEP (DSPCLIPPINGREGION NIL PANE)
                                     X Y)                                     ; Didn't end inside the window, abort cleanly
                      (\TEDIT.SHOWSEL SOURCESEL NIL)
                      (\TEDIT.SHOWSEL (FGETTOBJ TEXTOBJ SEL)
                            NIL)
                      (\TEDIT.SHOWSEL (FGETTOBJ TEXTOBJ SEL)
                            T)
                      (RETURN))
                  (CL:UNLESS (FGETTOBJ TEXTOBJ MENUFLG)                       ; Globals are documented, unfortunately.
                      (\TEDIT.SET.GLOBAL.SELECTIONS SELOPERATION SOURCESEL))
                  (CL:UNLESS (MEMB SELOPERATION '(NORMAL PENDINGDEL))
                      ;; If this is a normal selection, then the display now corresponds to SOURCESEL and is now correctly displayed.
                      ;; Otherwise, the selection in this TEXTOBJ is only a transient for this operation, turn it off here.

                      (\TEDIT.SHOWSEL SOURCESEL NIL))
             ;; Execute the SELOPERATION in the TTY process (maybe here)

                  (\TEDIT.DO.SELOPERATION SOURCESEL SELOPERATION TEXTOBJ PANE)
                  (CL:WHEN (AND (FGETSEL SOURCESEL SELOBJ)
                                (IMAGEOBJPROP (FGETSEL SOURCESEL SELOBJ)
                                       'WHENOPERATEDONFN))
                      (APPLY* (IMAGEOBJPROP (FGETSEL SOURCESEL SELOBJ)
                                     'WHENOPERATEDONFN)
                             (FGETSEL SOURCESEL SELOBJ)
                            PANE
                            'SELECTED SOURCESEL (FGETTOBJ TEXTOBJ STREAMHINT)))
                  (CL:WHEN SELFN                                    ; Maybe for logging of selections?
                      (APPLY* SELFN TEXTOBJ SOURCESEL SELOPERATION 'FINAL))]))])
```

(\TEDIT.DO.SELOPERATION

```
  [LAMBDA (SOURCESEL SELOPERATION TEXTOBJ PANE)                  ; Edited 21-Feb-2024 20:08 by rmk
                                                                 ; Edited 19-Feb-2024 00:13 by rmk

    ;; Executes SELOPERATION in the TTY process.  If the TTY process is a Tedit process (either this one or another one) and doesn't demand a
    ;; COPYINSERT, this is accomplished by setting variables in that process' command loop.Otherwise, does a COPYINSERT into the TTY.

    (LET* [(TTYPROC (TTY.PROCESS))
           (TTYW (PROCESSPROP TTYPROC 'WINDOW))
           (TTYTEXTOBJ (AND TTYW (fetch (TEXTWINDOW PTEXTOBJ) of TTYW]
          (CL:WHEN (AND TTYTEXTOBJ (OR (GETTEXTPROP TTYTEXTOBJ 'COPYBYBKSYSBUF)
                                       (FGETTOBJ TTYTEXTOBJ EDITOPACTIVE)))
                (SETQ TTYTEXTOBJ NIL))
          (SELECTQ SELOPERATION
              (COPY (CL:UNLESS TTYTEXTOBJ
                        (\TEDIT.COPYINSERT TTYW SOURCESEL)       ; Copy is done, nothing more to do
                        (SETQ SELOPERATION NIL)))
              (MOVE (CL:UNLESS TTYTEXTOBJ
                        (\TEDIT.COPYINSERT TTYW SOURCESEL)       ; Copy is done, have to delete source
                        (if T
                            then                                 ; A remaining mystery: we should be able to execute this in
                                                                 ; PANE's Tedit process
                                 (SETQ SELOPERATION NIL)
                                 (\TEDIT.DELETE TEXTOBJ SOURCESEL)
                          else (SETQ SELOPERATION 'DELETE))))
              (PENDINGDEL (FSETTOBJ TEXTOBJ BLUEPENDINGDELETE T)
                          (SETQ SELOPERATION 'NORMAL))
              NIL)
          (FSETTOBJ TEXTOBJ SELPANE PANE)
          (CL:WHEN (AND SELOPERATION TTYTEXTOBJ)

              ;; Order of variables matters:  SELOPERATION must be last.

              [PROCESS.EVAL TTYPROC `(PROGN (SETQQ SOURCESEL ,SOURCESEL)
                                            (SETQQ SELPANE ,PANE)
                                            (SETQQ SELOPERATION ,SELOPERATION])])
```

(\TEDIT.TTY.TEXTOBJP

```
  [LAMBDA NIL                                                    ; Edited  8-Feb-2024 16:52 by rmk

    ;; Returns the TEXTOBJ of the TTY process, if it is a TEDIT command-loop process, otherwise NIL.

    (LET* [(TTYPROC (TTY.PROCESS))
           (TTYW (PROCESSPROP TTYPROC 'WINDOW]
          (CL:WHEN TTYW
              (fetch (TEXTWINDOW PTEXTOBJ) of TTYW))])
```

(\TEDIT.BUTTONEVENTFN.SELOPERATION

```
  [LAMBDA (TEXTOBJ)                                              ; Edited 27-Jan-2024 12:55 by rmk
    (COND
       ((KEYDOWNP 'COPY)                                         ; In a read-only document, you can only copy.
        'COPY)
       ((AND (KEYDOWNP 'MOVE)
             (NOT (GETTOBJ TEXTOBJ TXTREADONLY)))                ; The MOVE key is down, so set MOVE mode.
        'MOVE)
       [(SHIFTDOWNP 'SHIFT)                                      ; the SHIFT key is down;  mark this selection for COPY or
                                                                 ; MOVE.
        (COND
           ((AND (SHIFTDOWNP 'CTRL)
                 (NOT (GETTOBJ TEXTOBJ TXTREADONLY)))            ; CTRL-SHIFT select means MOVE.
            'MOVE)
           (T 'COPY]
       ((SHIFTDOWNP 'META)                                       ; He's holding the meta key down , do a copylooks selection
        'COPYLOOKS)
       ((AND (SHIFTDOWNP 'CTRL)
             (NOT (GETTOBJ TEXTOBJ TXTREADONLY)))                ; Note that he's holding the control key down.
        'DELETE)
       (T 'NORMAL])
```

```
(\TEDIT.BUTTONEVENTFN.INACTIVE
  [LAMBDA (TEXTOBJ PANE)                                   ; 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 (GETTOBJ TEXTOBJ STREAMHINT)
                     PANE)
              NIL])

(\TEDIT.BUTTONEVENTFN.INTITLE
  [LAMBDA (Y PANE TEXTOBJ)                                 ; Edited 27-Jan-2024 10:42 by rmk

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

     (LET ((PREG (DSPCLIPPINGREGION NIL PANE))
           USERFN)
          (CL:WHEN (IGREATERP Y (fetch TOP of PREG))
              [COND
                 ((\TEDIT.MOUSESTATE RIGHT)
                  (DOWINDOWCOM PANE))
                 ((AND (OR (SHIFTDOWNP 'SHIFT)
                           (KEYDOWNP 'COPY))
                       (MOUSESTATE LEFT))
                  (bind THING unless (OR (SHIFTDOWNP 'SHIFT)
                                         (KEYDOWNP 'COPY))
                     do (GETMOUSESTATE)
                        (CL:UNLESS (INSIDEP PREG (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)))
                 ((MOUSESTATE (OR LEFT MIDDLE))
                  (CL:WHEN (AND (SETQ USERFN (WINDOWPROP PANE 'TEDIT.TITLEMENUFN))
                                (NEQ USERFN 'DON'T))
                      (ADD.PROCESS (LIST USERFN (KWOTE PANE))))))]
              T)])

(\TEDIT.COPYINSERT
  [LAMBDA (TTYW SOURCESEL)                                 ; Edited 17-Feb-2024 12:52 by rmk

     ;; Inserts the information in SOURCESEL into the TTY window.

     (if (AND NIL (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 (SOURCETOBJ _ (GETSEL SOURCESEL SELTEXTOBJ)) from (FGETSEL SOURCESEL CH#)
               to (SUB1 (FGETSEL SOURCESEL CHLIM)) while (SETQ CH (TEDIT.NTHCHARCODE SOURCETOBJ CHNO))
               do (CL:IF (IMAGEOBJP CH)
                      (COPYINSERT CH)
                      (BKSYSBUF (CHARACTER CH)))])
```

```
)

(MOVD? 'NILL '\TEDIT.COPYINSERT)

(DEFINEQ

(\TEDIT.PANE.SPLIT
  [LAMBDA (TEXTOBJ WINDOWTOSPLIT)                                    ; 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))
                      (SHOULDNT)))
                (T (CURSOR T]
        T)])

(\TEDIT.SPLITW
  [LAMBDA (OLDPANE Y)                                                ; Edited 20-Mar-2024 11:01 by rmk
                                                                     ; Edited 15-Mar-2024 22:00 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  1-Oct-2023 11:58 by rmk
                                                                     ; Edited 22-Sep-2023 20:53 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.

    ;; 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))
          (TEXTOBJ (TEXTOBJ! (fetch (TEXTWINDOW WTEXTOBJ) of OLDPANE)))
          (NEXTPANE (fetch (TEXTWINDOW NEXTPANE) of OLDPANE))
          ATTACHEDWINDOWS NEWPANE PROPS NEWFIRSTLINE SEL)
         (CL:UNLESS Y
             (SETQ Y (LASTMOUSEY OLDPANE)))                          ; Get the Y-position where we're to make the split--it's either
                                                                     ; supplied or we use the mouse's Y position.
         (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))
         (SHAPEW OLDPANE (create REGION using WREG BOTTOM _ (IPLUS (fetch BOTTOM of WREG)
                                                                   Y)
                                           HEIGHT _ (IDIFFERENCE (fetch HEIGHT of WREG)
                                                            Y))))    ; Reshape the original window to form the upper pane. This
                                                                     ; fixes/displays the current selection.

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

         (ATTACHWINDOW (SETQ NEWPANE (CREATEW (create REGION using WREG HEIGHT _ Y)
                                           NIL NIL NIL))
                 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 (fetch (CHARLOOKS CLFONT) of (FGETTOBJ TEXTOBJ CARETLOOKS))
                 NEWPANE)                                        ; Set the font on the display stream to be the current one from
                                                                     ; CARETLOOKS

         ;; Not sure if same PROPS as for PANE (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 (FGETTOBJ TEXTOBJ STREAMHINT)
                 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.

         (for S in (\TEDIT.COLLECTSELS TEXTOBJ) do (for PANE inpanes (PROGN TEXTOBJ) as L1
                                                      on (GETSEL S L1) as LN on (GETSEL S LN)
                                                      when (EQ PANE OLDPANE) do (push (CDR L1)
                                                                                      NIL)
                                                                                (push (CDR LN)
                                                                                      NIL)))

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

         [SETQ NEWFIRSTLINE (for L inlines (fetch (TEXTWINDOW PLINES) of OLDPANE) unless (FGETLD L NEXTLINE)
                              do (RETURN (\TEDIT.FORMATLINE TEXTOBJ (ADD1 (FGETLD L LCHARLIM]
         (\TEDIT.WINDOW.SETUP NEWPANE (FGETTOBJ TEXTOBJ STREAMHINT)
                 PROPS OLDPANE NEWFIRSTLINE)
         (CL:WHEN NEWFIRSTLINE
             (\TEDIT.FILLPANE (GETLD NEWFIRSTLINE PREVLINE)
                    TEXTOBJ NEWPANE))
         (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
         (\TEDIT.FIXSEL SEL TEXTOBJ NIL NEWPANE)
         (CL:WHEN (GETSEL SEL ONFLG)
             (SETSEL SEL ONFLG NIL)                              ; Turn it off, so we can turn it on for NEWPANE
             (\TEDIT.SHOWSEL SEL T NEWPANE))
         (WINDOWPROP NEWPANE 'PROCESS (WINDOWPROP OLDPANE 'PROCESS))
         (replace (TEXTWINDOW NEXTPANE) of OLDPANE with NEWPANE)    ; Tell the this pane about the new pane just 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)
                                                                     ; Tell the lower pane about its lower, lower pane..
             (replace (TEXTWINDOW NEXTPANE) of NEWPANE with NEXTPANE))])
```

(\TEDIT.UNSPLITW

```
  [LAMBDA (PANE)                                                ; Edited 20-Mar-2024 11:01 by rmk
                                                                ; Edited 15-Mar-2024 18:30 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* ((TEXTOBJ (TEXTOBJ! (fetch (TEXTWINDOW WTEXTOBJ) of PANE)))
            (PANES (GETTOBJ TEXTOBJ \WINDOW))
            (PRIMARYPANE (\TEDIT.MAINW PANE))
            (SEL (FGETTOBJ TEXTOBJ SEL))
           PRIORPANE NEXTPANE ATTACHEDWINDOWS)
          (CL:WHEN (EQ PANE PRIMARYPANE)
              (TEDIT.PROMPTPRINT TEXTOBJ "Can't UNSPLIT the main window" T)
              (RETURN))
          [SETQ PRIORPANE (find P in PANES suchthat (EQ PANE (fetch (TEXTWINDOW NEXTPANE) of P]
          (SETQ NEXTPANE (fetch (TEXTWINDOW NEXTPANE) of PANE))
          (\TEDIT.SHOWSEL SEL NIL)                              ; Turn off selections during the unsplit.
          (FSETTOBJ TEXTOBJ SELPANE PRIMARYPANE)
          (for P in PANES as CARET in (GETTOBJ TEXTOBJ CARET) as SL1 in (GETSEL SEL L1) as SLN
             in (GETSEL SEL LN) when (EQ PANE P) do (change (GETTOBJ TEXTOBJ CARET TEXTOBJ)
                                                           (DREMOVE CARET DATUM))
                                                    (change (GETSEL SEL L1)
                                                           (DREMOVE SL1 DATUM))
                                                    (change (GETSEL SEL LN)
                                                           (DREMOVE SLN DATUM))
                                                    (RETURN))
          (WINDOWPROP PANE 'CURSOROUTFN NIL)
          (WINDOWPROP PANE 'CURSORMOVEDFN NIL)                  ; Disconnect
          (replace (TEXTWINDOW WTEXTSTREAM) of PANE with NIL)
          (SETTOBJ TEXTOBJ \WINDOW (DREMOVE PANE PANES))
          (replace (TEXTWINDOW NEXTPANE) of PANE with NIL)
          (\TEDIT.FIXSEL (TEXTSEL TEXTOBJ)
                     TEXTOBJ)
              (replace (TEXTWINDOW NEXTPANE) of PRIORPANE with NEXTPANE)
        ;;
        ;; 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 PRIORPANE 'ATTACHEDWINDOWS NIL))
              [SHAPEW PRIORPANE (UNIONREGIONS (WINDOWPROP PANE 'REGION)
                                        (WINDOWPROP PRIORPANE 'REGION]
              (WINDOWPROP PRIORPANE 'ATTACHEDWINDOWS ATTACHEDWINDOWS)
              (CL:WHEN NEXTPANE
                  ;; PANE had a yet lower pane attached to it.  Promote it to PANE's position in the NEXTPANE chain
                  (DETACHWINDOW NEXTPANE)
                  (ATTACHWINDOW NEXTPANE PRIORPANE 'BOTTOM 'JUSTIFY 'MAIN))
              (CLOSEW PANE)
              (\TEDIT.SHOWSEL SEL T])
)

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

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

(RPAQ? \TEDIT.OP.WIDTH 12)

(RPAQ? \TEDIT.OP.BOTTOM 12)

(RPAQ? \TEDIT.LINEREGION.WIDTH 8)

(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 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 window. The test is that the returned window has a TEXTOBJ
       ;; property, and the TEXTOBJ thinks this is its window.
       (LET* [(TEXTOBJ (TEXTOBJ WINDOW T))
              (CHECKED-WINDOW (AND TEXTOBJ (\TEDIT.PRIMARYW TEXTOBJ]
             (AND CHECKED-WINDOW (MEMB CHECKED-WINDOW (FGETTOBJ TEXTOBJ \WINDOW))
                CHECKED-WINDOW])

)
```

;; 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 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))])

)
```

;; 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 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))
          PWINDOW MAINWINDOW)
         (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])
```

(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 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.
          [SETQ NEWTOP (fetch (REGION TOP) of (WINDOWPROP PROMPT-WINDOW 'REGION]
          [for WINDOW 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  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.PRIMARYW 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 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)
         (\TEDIT.WINDOW.TITLE TEXTOBJ NIL (\TEDIT.DEFAULT.TITLE (OR FILENAME TITLE)))
         (SETQ MENUSTREAM (TEDITMENU.STREAM TEXTOBJ))
         (CL:WHEN (AND MENUSTREAM (type? LITATOM TITLE))          ; if we have a filename then put it in the GET and PUT fields of
                                                                  ; the menu
             (SETQ FILENAME (PACKFILENAME 'VERSION NIL 'BODY TITLE))
             (MBUTTON.SET.FIELD MENUSTREAM 'Get FILENAME)
             (MBUTTON.SET.FIELD MENUSTREAM 'Put FILENAME))])
)
```

;; Screen updating utilities

```
(DEFINEQ
```

(TEDIT.DEACTIVATE.WINDOW

```
  [LAMBDA (W)                                                     ; Edited 20-Mar-2024 11:02 by rmk
                                                                  ; Edited 15-Mar-2024 13:34 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:
```

;; 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 for to unsplit panes. The actual window-clsoing 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 ((TEXTOBJ (TEXTOBJ W T)))
          (CL:UNLESS TEXTOBJ                                      ; Return NIL if not an editing window (rather than error?)
              (RETURN))
          (TEXTOBJ! TEXTOBJ)
      ;; Return DON'T If we don't close the window.  if previously quit, the window is closed already, and would be reopened to reclose it.
          (CL:WHEN (GETTOBJ TEXTOBJ EDITOPACTIVE)
              ;; If something is going on, DON'T CLOSE THE WINDOW
              (TEDIT.PROMPTPRINT TEXTOBJ "Not closed; edit operation in progress" T)
              (RETURN 'DON'T))
          (CL:UNLESS (\TEDIT.QUIT W T)
              (RETURN 'DON'T))
          (SETTOBJ TEXTOBJ EDITFINISHEDFLG T)                    ; This causes the command loop to return to \TEDIT1, where the
                                                                 ; closing actually happens
          (CL:WHEN (AND (GETTOBJ TEXTOBJ PROMPTWINDOW)
                        (OPENWP (GETTOBJ TEXTOBJ PROMPTWINDOW)))
              (CLEARW (GETTOBJ TEXTOBJ PROMPTWINDOW)))
          (\TEDIT.SHOWSEL (TEXTSEL TEXTOBJ)
                 NIL)                                            ; 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 W)
                        (EQ W (\TEDIT.MAINW TEXTOBJ)))           ; Reset the window's title to a known 'inactive' value
              (\TEDIT.WINDOW.TITLE TEXTOBJ "Edit Window [Inactive]"))
          (for PANE in (REVERSE (CDR (GETTOBJ TEXTOBJ \WINDOW))) do
                                          ;; Run thru any split-off sub-panes, and reattach them, so we get a whole window back before the end
                                          ;; of the world. Presumably we run through backwards because it looks better if the windows close
                                          ;; from the bottom up.
                                                                      (\TEDIT.UNSPLITW PANE))
          (SETTOBJ TEXTOBJ \WINDOW NIL)
          (CL:WHEN (type? STREAM (GETTOBJ TEXTOBJ TXTFILE))      ; Close the file that this window was open on.
              (CL:UNLESS (fetch (TEXTWINDOW CLOSINGFILE) of W)
                  (replace (TEXTWINDOW CLOSINGFILE) of W with T)
                  (CLOSEF? (GETTOBJ TEXTOBJ TXTFILE))))
          (replace (TEXTWINDOW WLINES) of W with NIL)
          (WINDOWPROP W 'PROCESS.EXITFN NIL)
          (WINDOWPROP W 'PROCESS.IDLEFN NIL)
          (WINDOWPROP W 'BUTTONEVENTFN 'TOTOPW)                  ; And the button functions
          (WINDOWPROP W 'RIGHTBUTTONFN 'DOWINDOWCOM)
          (WINDOWDELPROP W 'CLOSEFN 'TEDIT.DEACTIVATE.WINDOW)    ; To avoid a loop
          (WINDOWPROP W 'SCROLLFN NIL)
          (WINDOWDELPROP W 'RESHAPEFN '\EDITRESHAPEFN)
          (\TEDIT.INTERRUPT.SETUP (WINDOWPROP W 'PROCESS)
                 T)                                              ; Restore any disarmed interrupts.
          (for MENUW in (ATTACHEDWINDOWS W) 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 W)                                   ; Now kill this one
          (DISMISS 300)
          [SETTOBJ TEXTOBJ \WINDOW (CL:WHEN (LISTP (GETTOBJ TEXTOBJ \WINDOW))
                                                                 ; It's a list;  remove this pane
                                       (DREMOVE W (GETTOBJ TEXTOBJ \WINDOW)))]
          (WINDOWPROP W 'CURSOROUTFN NIL)
          (WINDOWPROP W 'CURSORMOVEDFN NIL)                      ; Disconnect
          (replace (TEXTWINDOW WTEXTSTREAM) of W with NIL])
```

(\TEDIT.REPAINTFN

```
  [LAMBDA (PANE)                                                 ; Edited 20-Mar-2024 06:43 by rmk
                                                                 ; Edited 15-Mar-2024 13:36 by rmk
                                                                 ; Edited 13-Dec-2023 23:27 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
    ;; If PANE is a pane of a split window, all sister panes will be refreshed, in keeping with the illusion that PANE is one part of a larger "window".
    (LET ((TEXTOBJ (fetch (TEXTWINDOW PTEXTOBJ) of PANE))
          SEL WASON)
         (CL:WHEN TEXTOBJ
             (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
             (SETQ WASON (AND (GETSEL SEL SET)
                              (GETSEL SEL ONFLG)))
             ;; The window is clear before this is called, so no highlighting to worry about.  Tell the selection.
             (FSETSEL SEL ONFLG NIL)
             (for P PLINES inpanes (PROGN TEXTOBJ) do (\TEDIT.LINES.BELOW (fetch (TEXTWINDOW PLINES) of P)
                                                             NIL P TEXTOBJ))
             (CL:WHEN WASON
                 (\TEDIT.FIXSEL SEL TEXTOBJ)                     ; Account for new lines and highlighting
                 (\TEDIT.SHOWSEL SEL T))))])

(\TEDIT.AFTERMOVEFN
  [LAMBDA (PANE)                                                      ; 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.
                                                                      ; Edited 17-Dec-2023 17:31 by rmk
    (CL:WHEN (WINDOWPROP PANE 'OFFSCREEN)
           (\TEDIT.REPAINTFN PANE))
    (WINDOWPROP PANE 'OFFSCREEN (OFFSCREENP PANE])

(OFFSCREENP
  [LAMBDA (WINDOW)                                                    ; 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 LEFT of REGION)
                          0)
             (PUSH RESULT 'LEFT))
         (CL:WHEN (IGREATERP (fetch 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 TOP of REGION)
                          (fetch (SCREEN SCHEIGHT) of SCREEN))
             (PUSH RESULT 'TOP))
         (CL:WHEN (ILESSP (fetch BOTTOM of REGION)
                          0)
             (PUSH RESULT 'BOTTOM))
         (CL:WHEN (OR (MEMB 'TOP RESULT)
                      (MEMB 'BOTTOM RESULT))
             (PUSH RESULT 'VERTICAL))
         RESULT])

(\TEDIT.RESHAPEFN
  [LAMBDA (PANE BITS OLDREGION)                                       ; Edited 20-Mar-2024 06:46 by rmk
                                                                      ; Edited 16-Mar-2024 00:02 by rmk
                                                                      ; Edited 20-Jan-2024 23:02 by rmk
                                                                      ; Edited  2-Jan-2024 12:43 by rmk
                                                                      ; Edited 14-Dec-2023 11:32 by rmk
                                                                      ; Edited 20-Nov-2023 11:04 by rmk
                                                                      ; Edited  3-Nov-2023 12:10 by rmk
                                                                      ; Edited 11-May-2023 00:39 by rmk
                                                                      ; Edited 18-Apr-2023 23:46 by rmk
                                                                      ; Edited  5-Apr-2023 09:23 by rmk
                                                                      ; Edited 30-May-91 23:34 by jds

    ;; Will eventually do the right thing w/r/t text margins.  For now, it's a place holder.

    ;; This is called after the window has been reshaped, we have to redisplay

    (PROG ((TEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of PANE))
           (PREG (DSPCLIPPINGREGION NIL PANE))
           NEWPHEIGHT PLINES LINE)
          (CL:UNLESS TEXTOBJ                                          ; Not a Tedit window
              (RETURN))
          (CL:UNLESS (SETQ PLINES (fetch (TEXTWINDOW PLINES) of PANE))

              ;; Should always be a dummy line, but...

              (RETURN))
          (\TEDIT.SHOWSEL (FGETTOBJ TEXTOBJ SEL)
                 NIL)                                                 ; Turn off the selection while we make changes
          (SETQ NEWPHEIGHT (fetch HEIGHT of PREG))
          (FSETTOBJ TEXTOBJ WTOP NEWPHEIGHT)                          ; Save new height/width for later use
          (FSETTOBJ TEXTOBJ WRIGHT (fetch WIDTH of PREG))
          (FSETTOBJ TEXTOBJ WBOTTOM (fetch BOTTOM of PREG))
          (FSETTOBJ TEXTOBJ WLEFT (fetch LEFT of PREG))

     ;; Hunt for the first line that had been visible, so we can find the CH# that has to appear at the top of the pane.  THIS SHOULD JUST BE
     ;; (NEXTLINE PLINES), IF WE ALWAYS FLUSH UNSEEN LINES.

          (SETQ LINE (find L (OLDPHEIGHT _ (fetch HEIGHT of OLDREGION))
                        inlines
                        (GETLD PLINES NEXTLINE) suchthat (ILESSP (FGETLD L LTRUEYTOP)
                                                                  OLDPHEIGHT)))
          (SETLD PLINES YBOT NEWPHEIGHT)
          (CL:WHEN LINE                                               ; If nothing visible then, nothing now
              (CL:WHEN (IGREATERP (FGETTOBJ TEXTOBJ TEXTLEN)
                                  0)
                  [SETQ LINE (CADR (\TEDIT.LINES.ABOVE TEXTOBJ (FGETLD LINE LCHAR1])
              (CL:UNLESS (EQ LINE PLINES)                                ; Forget the old chain of line descriptors
                  (LINKLD PLINES LINE))                                  ; Fix the line to appear at the top of the pane
              (SETYPOS LINE (IDIFFERENCE NEWPHEIGHT (FGETLD LINE LHEIGHT)))
              (\TEDIT.DISPLAYLINE TEXTOBJ LINE PANE)                     ; Actually display it
              (\TEDIT.FILLPANE LINE TEXTOBJ PANE))
          (\TEDIT.FIXSEL (FGETTOBJ TEXTOBJ SEL)
                 TEXTOBJ)                                                ; Fix up the selection to account for the line shuffling
          (\TEDIT.SHOWSEL (FGETTOBJ TEXTOBJ SEL)
                 T])
```

(\TEDIT.PANEWITHINSCREEN?

```
  [LAMBDA (PANE)                                                          ; Edited 20-Nov-2023 13:43 by rmk
                                                                          ; Edited 10-May-2023 23:37 by rmk
```

;; True if PANE is completely within the screen and therefore that it is safe to reuse image-bits that were previously displayed anywhere within
;; PANE's clipping region.

;; \TEDIT.AFTERMOVEFN should record this as a property on the PANE. Also, if this is false after a move, then the aftermovefn should force a
;; redisplay.

;; Since TEDIT doesn't (yet) support horizontal scrolling), we only test the vertical dimension

;;

```
    (LET [(PANEREG (WINDOWPROP PANE 'REGION]
         (AND (IGEQ (fetch (REGION BOTTOM) of PANEREG)
                    0)
              (OR T (IGEQ (fetch (REGION LEFT) of PANEREG)
                          0))
              (ILEQ (fetch (REGION PTOP) of PANEREG)
                    SCREENHEIGHT)
              (OR T (ILEQ (fetch (REGION PRIGHT) of PANEREG)
                          SCREENWIDTH])

)

(DEFINEQ
```

(\TEDIT.SCROLLFN

```
  [LAMBDA (PANE DX DY)
```

;; 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 [(TEXTOBJ (TEXTOBJ! (fetch (TEXTWINDOW PTEXTOBJ) of PANE]
          (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 (FGETTOBJ TEXTOBJ STREAMHINT)
                            '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.SCROLLFLOAT TEXTOBJ PANE DY)
            elseif (IGREATERP DY 0)
              then (\TEDIT.SCROLLUP TEXTOBJ PANE DY)
            elseif (ILESSP DY 0)
              then (\TEDIT.SCROLLDOWN TEXTOBJ PANE DY))
          (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ PANE)
          (CL:WHEN (GETTEXTPROP TEXTOBJ 'POSTSCROLLFN)                   ; For user subsystem cleanup
              (DOUSERFNS (GETTEXTPROP TEXTOBJ 'POSTSCROLLFN)
                     PANE)))
    NIL])
```

(\TEDIT.SCROLLFLOAT

```
  [LAMBDA (TEXTOBJ PANE DY)                                               ; Edited 20-Mar-2024 06:46 by rmk
                                                                          ; Edited 15-Mar-2024 22:00 by rmk
                                                                          ; Edited 22-Jan-2024 10:43 by rmk
                                                                          ; Edited 20-Jan-2024 23:13 by rmk
                                                                          ; Edited  2-Jan-2024 11:02 by rmk
                                                                          ; Edited 13-Dec-2023 23:24 by rmk
                                                                          ; Edited  4-Dec-2023 11:25 by rmk
```

```
                                                                ; Edited 28-Nov-2023 12:10 by rmk
                                                                ; Edited 24-Nov-2023 13:00 by rmk
                                                                ; Edited 22-Nov-2023 14:42 by rmk
                                                                ; Edited 20-Nov-2023 14:19 by rmk
                                                                ; Edited  3-Nov-2023 12:10 by rmk
                                                                ; Edited 30-Mar-2023 23:38 by rmk

    ;; Thumb scrolling, DY is FLOATP.

    (LET ((CH# (IMAX [IMIN (SUB1 (TEXTLEN TEXTOBJ))
                          (FIXR (FTIMES DY (TEXTLEN TEXTOBJ]
                     1))
          (PREG (DSPCLIPPINGREGION NIL PANE))
          (SEL (FGETTOBJ TEXTOBJ SEL))
          PHEIGHT NEWTOP)
         (SETQ PHEIGHT (fetch HEIGHT of PREG))                  ; Height of the pane

         ;; Does any currently formatted  line include the target char? This will become the new top line

         (SETQ NEWTOP (find L inlines (GETLD (fetch (TEXTWINDOW PLINES) of PANE)
                                           NEXTLINE)
                         suchthat (WITHINLINEP CH# L)))
         (COND
            (NEWTOP

                   ;; If so, convert to an integer scroll so the screen is not blanked and reformatted unnecessarily

                   [SETQ DY (COND
                               [(ILEQ PHEIGHT (GETLD NEWTOP YBOT))
                                                                ; NEWTOP is off the top of the window
                                (IMINUS (for L inlines (GETLD NEWTOP NEXTLINE)
                                           while (ILEQ PHEIGHT (FGETLD L YBOT)) sum
                                                                          ;; sum the heights of all lines between the
                                                                          ;; NEWTOP and the present top line

                                                                                  (FGETLD L LHEIGHT]
                               (T                               ; NEWTOP is in the window or below, raise it up
                                  (IDIFFERENCE (IDIFFERENCE PHEIGHT (GETLD NEWTOP YBOT))
                                         (GETLD NEWTOP LHEIGHT]
                   (if (IGREATERP DY 0)
                       then (\TEDIT.SCROLLUP TEXTOBJ PANE DY)
                     elseif (ILESSP DY 0)
                       then (\TEDIT.SCROLLDOWN TEXTOBJ PANE DY)))
            (T ;; There is no current line to be moved to the top of the pane. The line containing CH# becomes the new topline of the pane

               [SETQ NEWTOP (CADR (\TEDIT.LINES.ABOVE TEXTOBJ CH# (fetch BOTTOM of PREG]
               (SETYPOS NEWTOP (IDIFFERENCE PHEIGHT (GETLD NEWTOP LHEIGHT)))
               (LINKLD (fetch (TEXTWINDOW PLINES) of PANE)
                      NEWTOP)                                   ; New block lines replace previous pane lines
               (\TEDIT.CLEARPANE PANE)

               ;; Maybe replace the rest of this with \TEDIT.REPAINTFN ?  \FILLPANE adds the dummy lines

               (\TEDIT.FILLPANE (fetch (TEXTWINDOW PLINES) of PANE)
                      TEXTOBJ PANE)
               (\TEDIT.FIXSEL SEL TEXTOBJ NIL PANE)
               (CL:WHEN (FGETSEL SEL ONFLG)

                   ;; Tell the selection that none of its hilighting is current onscreen (BLTSHADE above), then restore it.

                   (FSETSEL SEL ONFLG NIL)
                   (\TEDIT.SHOWSEL SEL T PANE))])
```

(\TEDIT.SCROLLUP

```
  [LAMBDA (TEXTOBJ PANE DY)                                     ; Edited 20-Mar-2024 06:43 by rmk
                                                                ; Edited 15-Mar-2024 19:23 by rmk
                                                                ; Edited 14-Dec-2023 00:00 by rmk
                                                                ; Edited  4-Dec-2023 20:49 by rmk
                                                                ; Edited 30-Nov-2023 00:07 by rmk
                                                                ; Edited 28-Nov-2023 22:55 by rmk
                                                                ; Edited 28-Apr-2023 08:55 by rmk
                                                                ; Edited 24-Apr-2023 23:48 by rmk

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

    (PROG ((PREG (DSPCLIPPINGREGION NIL PANE))
           (TEXTLEN (FGETTOBJ TEXTOBJ TEXTLEN))
           PBOTTOM PHEIGHT PLINES OLDTOPLINE NEWTOPLINE LASTVISIBLE WHERESEL DELTA)
          (SETQ PHEIGHT (fetch HEIGHT of PREG))                 ; Height of the pane
          (SETQ PBOTTOM (\TEDIT.ONSCREEN? PANE 'BOTTOM))        ; Effective bottom
          (SETQ PLINES (fetch (TEXTWINDOW PLINES) of PANE))
          (CL:WHEN (IGREATERP (GETLD (GETLD PLINES NEXTLINE)
                                     LCHARLIM)
                          TEXTLEN)
              (HELP 'TOP))                                      ; Currently formatted PANE lines

     ;; Find the first line at least DY below the top of the pane.

     ;; The initial scan from PLINES is needed in case invisible lines have been kept in the chain above the current top line.

          (SETQ OLDTOPLINE (find L inlines (GETLD PLINES NEXTLINE) suchthat (ILEQ (FGETLD L YTOP)
                                                                                  PHEIGHT)))
          (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))
     ;;
          (SETQ WHERESEL (\TEDIT.WHERE.SEL TEXTOBJ OLDTOPLINE PANE))
     ;;
     ;; Walk down a sequence of lines until we arrive at a line that is DY from the top. 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 (for L NEXT inlines OLDTOPLINE first (CL:WHEN NIL
                                                                  (ILESSP (IPLUS DY (FGETLD OLDTOPLINE LTRUEYTOP)
                                                                                )
                                                                         PHEIGHT)
                                    ;; If the old truetop would still be visible after raising DY, then move it up by DY.  This effectively
                                    ;; discounts the white space of paragraph leading.  Maybe we also want to discount the white space
                                    ;; of line-leading below, by using LTRUEHEIGHT instead of LHEIGHT to determine the new bottom
                                    ;; line
                                                                  (SETQ NEXT (\TEDIT.FORMATLINE TEXTOBJ
                                                                                    (FGETLD OLDTOPLINE LCHAR1)))
                                                                  (LINKLD NEXT (FGETLD OLDTOPLINE NEXTLINE))
                                                                  (SETYPOS NEXT (IPLUS DY (FGETLD OLDTOPLINE YBOT
                                                                                                 )))
                                                                  (RETURN NEXT))
                               do (add DY (IMINUS (FGETLD L LHEIGHT)))
                                  (CL:WHEN (OR (ILEQ DY 0)
                                               (IGEQ (FGETLD L LCHARLIM)
                                                     TEXTLEN))
                                         (RETURN L))
                                  (CL:UNLESS (FGETLD L NEXTLINE)
                                     ;; 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 line is positioned
                                     ;; properly 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 TEXTOBJ (ADD1 (FGETLD L LCHARLIM]
                                     (LINKLD L NEXT)                 ; So we find NEXT on the next iteration
                                     (\TEDIT.LINE.BOTTOM NEXT))]
          (CL:UNLESS NEWTOPLINE                                       ; If nothing found, nothing can be done
              (RETURN))
          (CL:WHEN (EQ OLDTOPLINE NEWTOPLINE)                         ; Move at least one line
              (SETQ NEWTOPLINE (FGETLD OLDTOPLINE NEXTLINE)))
          (CL:UNLESS (AND NEWTOPLINE (ILEQ (FGETLD NEWTOPLINE LCHARLIM)
                                           TEXTLEN))
                 (RETURN))
     ;;
     ;; NEWTOPLINE is good to go.
     ;; Lines above NEWTOPLINE are chopped off, it and all lines below must be repositioned to the top of PANE, and new lines must be created to fill
     ;; the space after the last visible line is raised.
          (LINKLD PLINES NEWTOPLINE)                                  ; Chop off lines above that are no longer visible
     ;;
          (CL:UNLESS (IGEQ (GETLD NEWTOPLINE YBOT)
                           PBOTTOM)                                   ; Not visible, SUB1 to display not quite at the top, then raise
              (\TEDIT.LINE.BOTTOM NEWTOPLINE)
              (\TEDIT.DISPLAYLINE TEXTOBJ NEWTOPLINE PANE))
     ;;
     ;; Raise NEWTOPLINE so that its top is at PHEIGHT, and reposition all lines below. DELTA presumably is (or is close to) the original DY.
          (SETQ DELTA (IDIFFERENCE PHEIGHT (FGETLD NEWTOPLINE YTOP)))
          (for L inlines NEWTOPLINE while (IGEQ (FGETLD L YBOT)
                                                PBOTTOM)
             do (SETYPOS L (IPLUS DELTA (FGETLD L YBOT)))
                (SETQ LASTVISIBLE L))
     ;;
     ;; The effective PBOTTOM is the bottom of the clipping region that is onscreen
          (BITBLT PANE 0 PBOTTOM PANE 0 (IPLUS PBOTTOM DELTA)
                 (fetch WIDTH of PREG)
                 (FGETLD NEWTOPLINE YTOP)
                 'INPUT
                 'REPLACE)
          (CL:WHEN LASTVISIBLE
              ;; Clear any garbage left over below the (repositioned) LASTVISIBLELINE, and display new lines needed to fill the space
              (BLTSHADE WHITESHADE PANE 0 0 (fetch WIDTH of PREG)
                     (FGETLD LASTVISIBLE YBOT)
                     'REPLACE)
              (\TEDIT.LINES.BELOW LASTVISIBLE NIL PANE TEXTOBJ))
          (\TEDIT.SCROLL.SHOWSEL 'UP WHERESEL PANE TEXTOBJ LASTVISIBLE])
```

```
(\TEDIT.SCROLL.SHOWSEL
  [LAMBDA (DIRECTION WHERESEL PANE TEXTOBJ VISIBLELINE)          ; Edited 15-Mar-2024 13:36 by rmk
                                                                 ; Edited 18-Feb-2024 15:24 by rmk
                                                                 ; Edited 28-Nov-2023 22:58 by rmk

    ;; This synchronizes the selection hilighting and caret to correspond to lines that up- or down-scrolling has newly revealed.  It assumes that the
    ;; hilighting of previously and still visible lines have been carried along with BITBLT.  VISIBLELINE is the last visible line (above or below) that
    ;; bounds the region that needs to be redisplayed.

    (CL:WHEN WHERESEL                                            ; SEL was on, but now off.
        (LET ((SEL (FGETTOBJ TEXTOBJ SEL)))
             (if (OR (EQ 'BELOW (CAR WHERESEL))
                     (EQ 'ABOVE (CDR WHERESEL)))
                 then ;; No lines previously visible, they might come from above or below, in the ordinary way.

                      (FSETSEL SEL ONFLG NIL)
                      (\TEDIT.FIXSEL SEL TEXTOBJ NIL PANE)
                      (\TEDIT.SHOWSEL SEL T PANE)
               else ;; Old lines were visible and therefore highlighted, newly revealed lines may need to catch up.

                    (FSETSEL SEL ONFLG T)
                    (\TEDIT.FIXSEL SEL TEXTOBJ NIL PANE)
                    (SELECTQ DIRECTION
                        (UP ;; lastline is not highlighted when it comes up

                            (CL:WHEN (AND (EQ 'BELOW (CDR WHERESEL))
                                          VISIBLELINE
                                          (FGETLD VISIBLELINE NEXTLINE)
                                          (\TEDIT.SEL.LN SEL PANE TEXTOBJ))
                                (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ (FGETLD VISIBLELINE NEXTLINE)
                                       (\TEDIT.SEL.LN SEL PANE TEXTOBJ)
                                       PANE SEL)))
                        (DOWN                                    ; First line is not highlighted when it comes down
                              (CL:WHEN (EQ 'ABOVE (CAR WHERESEL))
                                  (\TEDIT.SHOWSEL.HILIGHT TEXTOBJ (\TEDIT.SEL.L1 SEL PANE TEXTOBJ)
                                         (FGETLD VISIBLELINE PREVLINE)
                                         PANE SEL)))
                        (SHOULDNT))
                    (\TEDIT.SETCARET SEL PANE TEXTOBJ T))))])

(\TEDIT.SCROLLDOWN
  [LAMBDA (TEXTOBJ PANE DY)                                      ; Edited 19-Mar-2024 23:34 by rmk
                                                                 ; Edited 15-Mar-2024 22:02 by rmk
                                                                 ; Edited 20-Jan-2024 23:13 by rmk
                                                                 ; Edited  2-Jan-2024 00:25 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
                                                                 ; Edited  3-Apr-2023 10:00 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, and resulting garbage at the bottom
    ;; of the window is cleared, and then the newlines are displayed to fill the space at the top.

    ;;

    (SETQ DY (IMINUS DY))                                        ; Now positive
    (PROG ((PREG (DSPCLIPPINGREGION NIL PANE))
           (VBOTTOM (\TEDIT.ONSCREEN? PANE 'BOTTOM))
           (PLINES (fetch (TEXTWINDOW PLINES) of PANE))
           PHEIGHT PBOTTOM OLDTOPLINE TOPOFOLD NEWTOPLINE WHERESEL LASTVISIBLE)
          (SETQ PHEIGHT (fetch HEIGHT of PREG))                  ; Height of the pane. Presumably there are no old lines above it,
                                                                 ; but...
          (SETQ PBOTTOM (fetch BOTTOM of PREG))
          (SETQ OLDTOPLINE (find L inlines (GETLD PLINES NEXTLINE) suchthat (ILESSP (FGETLD L YBOT)
                                                                                    PHEIGHT)))

     ;; Look backwards from the line before OLDTOP. IMAX and HEIGHT+1 to scroll at least one line

          (CL:WHEN (AND OLDTOPLINE (ILEQ (GETLD OLDTOPLINE LCHAR1)
                                         1))
                 (RETURN))

     ;;

          (SETQ NEWTOPLINE (if OLDTOPLINE
                               then (SETQ TOPOFOLD (FGETLD OLDTOPLINE YTOP))
                                    (\TEDIT.BACKFORMAT TEXTOBJ (IMAX DY (ADD1 (FGETLD OLDTOPLINE LHEIGHT)))
                                           (SUB1 (FGETLD OLDTOPLINE LCHAR1))
                                           (FGETLD OLDTOPLINE LHEIGHT))
                             else ;; If we didn't find a visible line, we must be looking at the tail end of the text.  We will need to bring down
                                  ;; some of its final lines.

                                  (\TEDIT.BACKFORMAT TEXTOBJ DY (FGETTOBJ TEXTOBJ TEXTLEN)
                                         0)))
          (CL:UNLESS NEWTOPLINE (RETURN))
          (SETQ WHERESEL (\TEDIT.WHERE.SEL TEXTOBJ OLDTOPLINE PANE))

      ;; NEWTOPLINE is at least one new line in front of OLDTOPLINE. We concatenate them into a single chain with Ypositions adjusted so that the
      ;; top of NEWTOPLINE is at PHEIGHT.

          (SETQ LASTVISIBLE (\TEDIT.NCONC.LINES NEWTOPLINE OLDTOPLINE PHEIGHT PBOTTOM))
          (LINKLD PLINES NEWTOPLINE)

      ;; All the needed lines have been constructed, linked, and positioned; some trailing lines may have been chopped.  We now try to be clever and
      ;; flicker-free about updating the display, although at this point a simple repaint will give the proper display.

          (CL:UNLESS (\TEDIT.OFFSCREEN.SCROLL TEXTOBJ PANE 'TOP)
              (CL:WHEN (AND OLDTOPLINE (IGEQ (FGETLD OLDTOPLINE YBOT)
                                             (FGETLD LASTVISIBLE YBOT)))

                  ;; The images for at least OLDTOPLINE and any lines below are currently in the bitmap.  Move it down.

                  (BITBLT PANE 0 (IPLUS VBOTTOM (IDIFFERENCE TOPOFOLD (FGETLD OLDTOPLINE YTOP)))
                         PANE 0 VBOTTOM (fetch WIDTH of PREG)
                         PHEIGHT
                         'INPUT
                         'REPLACE))
              (CL:WHEN LASTVISIBLE                                  ; Clear out any cruft in the bitmap that lowering might have
                                                                    ; produced.
                  (BLTSHADE WHITESHADE PANE 0 PBOTTOM (fetch WIDTH of PREG)
                         (FGETLD LASTVISIBLE YBOT)
                         'REPLACE))

              ;; Display the new lines.  No need to clear the rectangle first, \DISPLAYLINE fills the bitmap

              (for L inlines NEWTOPLINE until (EQ L OLDTOPLINE) do (\TEDIT.DISPLAYLINE TEXTOBJ L PANE))
              (\TEDIT.SCROLL.SHOWSEL 'DOWN WHERESEL PANE TEXTOBJ OLDTOPLINE)))
    NIL])
```

(\TEDIT.OFFSCREEN.SCROLL

```
  [LAMBDA (TEXTOBJ PANE BOUNDARY)                                   ; Edited 19-Mar-2024 23:34 by rmk
                                                                    ; Edited 15-Mar-2024 22:00 by rmk
                                                                    ; Edited 20-Jan-2024 23:12 by rmk
                                                                    ; Edited  2-Jan-2024 12:36 by rmk
                                                                    ; Edited  4-Dec-2023 20:57 by rmk
                                                                    ; Edited 11-May-2023 11:35 by rmk
                                                                    ; Edited 30-May-91 23:34 by jds

    ;; Returns NIL if PANE if PANE is not all or partially offscreen because of a previous move. Otherwise, this replaces the normal incremental screen
    ;; update of the calling function. Essentially, it applies a version of the repaint function for the one offscreen PANE of TEXTOBJ that is being
    ;; scrolled.

    (CL:WHEN (EQMEMB BOUNDARY (WINDOWPROP PANE 'OFFSCREEN))
        (LET (SEL WASON PREVLINE)
             (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
             (CL:WHEN (SETQ WASON (GETSEL SEL ONFLG))
                    (\TEDIT.SHOWSEL SEL NIL))                       ; Turn off the selection while we make changes

             ;; Find the precursor of the target top line.

             [SETQ PREVLINE (for L (PHEIGHT _ (fetch HEIGHT of (DSPCLIPPINGREGION NIL PANE)))
                                 inlines
                                 (fetch (TEXTWINDOW PLINES) of PANE) when (ILESSP (FGETLD L YBOT)
                                                                                   PHEIGHT)
                               do (RETURN (FGETLD L PREVLINE)) finally (RETURN (fetch (TEXTWINDOW PLINES)
                                                                                    of PANE]
             (\TEDIT.CLEARPANE PANE)
             (\TEDIT.FILLPANE PREVLINE TEXTOBJ PANE)
             (CL:WHEN WASON
                 (\TEDIT.FIXSEL SEL TEXTOBJ)                        ; Account for any line shuffling and re-highlight
                 (\TEDIT.SHOWSEL SEL T)))
        T)])
```

(\TEDIT.WHERE.SEL

```
  [LAMBDA (TEXTOBJ TOPLINE PANE)                                    ; Edited 20-Jan-2024 22:17 by rmk
                                                                    ; Edited 28-Nov-2023 11:48 by rmk
                                                                    ; Edited 25-Nov-2023 15:47 by rmk
                                                                    ; Edited 24-Nov-2023 12:47 by rmk

    ;; Returns the position of the selection in PANE relative to the first visible line in PANE, TOPLINE.  NIL if the selection isn't set aand on.

    (CL:WHEN TOPLINE
        (LET (WHERESEL (SEL (FGETTOBJ TEXTOBJ SEL)))
             (CL:WHEN (AND (FGETSEL SEL SET)
                           (FGETSEL SEL ONFLG))
                 (SETQ WHERESEL (\TEDIT.WHERE.SEL1 SEL TOPLINE PANE))
                 (CL:UNLESS (OR (EQ 'BELOW (CAR WHERESEL))
                                (EQ 'ABOVE (CDR WHERESEL)))

                     ;; At least partly visible.  Only flush the caret, leave (perhaps partial) highlighting for BITBLT

                     (\TEDIT.SETCARET SEL PANE TEXTOBJ 'DISABLE))
                 WHERESEL))))])
```

(\TEDIT.WHERE.SEL1

```
  [LAMBDA (SEL TOPLINE PANE)                                        ; Edited 24-Nov-2023 12:49 by rmk
                                                                    ; Edited 18-Nov-2023 23:52 by rmk
```

```
;; Determines the relationship of the start end end of SEL to the lines that are visible PANE.  This assumes that TOPLINE is the first currently
;; visible line. We search PANE for the last selected line, since we don't trust LN.

;; The value is a pair (ABOVE/IN/BELOW, indicating whether the start/end of the selection is above, in, or below the pane.

;; This is used to do incremental highlighting that avoids flickering with wheel-scroll spinning.

(LET (LASTVISIBLE)

     ;; Our search ends either when we run off PANE or we encounter the last line of the selection.

     ;; If we stop at the last SEL line, we will determine that the it is not BELOW

     [SETQ LASTVISIBLE (for L (CHLAST _ (SUB1 (FGETSEL SEL CHLIM)))
                            (PBOTTOM _ (fetch BOTTOM of (DSPCLIPPINGREGION NIL PANE)))
                            inlines TOPLINE do (CL:WHEN (WITHINLINEP CHLAST L)
                                                   (RETURN L))
                                               (CL:WHEN (ILESSP (FGETLD L YBOT)
                                                               PBOTTOM)
                                                   (RETURN (OR $$PREVLINE TOPLINE)))
                          finally (RETURN (OR $$PREVLINE TOPLINE]

     ;; (Don't put comment between ifs--they go into the CONS)

     (CONS (if (ILESSP (FGETSEL SEL CH#)
                       (FGETLD TOPLINE LCHAR1))
               then ;; SEL begins above TOPLINE, the first visible line

                    'ABOVE
             elseif (IGREATERP (FGETSEL SEL CH#)
                            (FGETLD LASTVISIBLE LCHARLIM))
               then ;;

                    'BELOW
             else 'IN)
           (if (ILESSP (SUB1 (FGETSEL SEL CHLIM))
                       (FGETLD TOPLINE LCHAR1))
               then ;; SEL ends above TOPLINE (presumably SELBEGIN is also ABOVE)

                    'ABOVE
             elseif (IGREATERP (SUB1 (FGETSEL SEL CHLIM))
                            (FGETLD LASTVISIBLE LCHARLIM))
               then ;; Not above, not displayed: must be below.

                    'BELOW
             else 'IN])

)

(DEFINEQ
```

(\TEDIT.ONSCREEN

```
  [LAMBDA (PANE)                                                ; Edited 19-Nov-2023 20:23 by rmk

    ;; If the clipping region is entirely onscreen, this returns the BOTTOM and TOP of the clipping region.

    ;; If it is off the screen at the bottom or top, this returns the effective bottom and top, in clipping region coordinates, of the part of the region that is
    ;; visible on the screen.

    ;; This assumes that the clipping region is in the window's coordinate system.

    (LET ((PREG (DSPCLIPPINGREGION NIL PANE))
          (YOFFSET (DSPYOFFSET NIL PANE)))
         (LIST (if (IGEQ YOFFSET 0)
                   then ;; Bottom is not below screen, could be above the top

                        (fetch BOTTOM of PREG)
                 else ;; Bottom is below screen

                      (IDIFFERENCE (fetch BOTTOM of PREG)
                             YOFFSET))
               (if (ILEQ (IPLUS YOFFSET (fetch TOP of PREG))
                         SCREENHEIGHT)
                   then ;; Top is not above screen, could be below the bottom

                        (fetch TOP of PREG)
                 else ;; Top is above screen

                      (IDIFFERENCE SCREENHEIGHT YOFFSET])
```

(\TEDIT.ONSCREEN?

```
  [LAMBDA (PANE PROP)                                           ; 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 the clipping region that is still visible.  E.g. if the bottom is below the screen by N
    ;; points (REG's bottom is -N), the return will be the clipping regions bottom _N.

    (LET [VAL (PREG (DSPCLIPPINGREGION NIL PANE))
              (BORDER (OR 0 (WINDOWPROP PANE 'BORDER]
         (SELECTQ PROP
             (BOTTOM (SETQ VAL (fetch (REGION BOTTOM) of (fetch REG of PANE)))
                     (if (ILESSP VAL 0)
                          then (IPLUS BORDER (IDIFFERENCE (fetch BOTTOM of PREG)
                                                         VAL))
                        else (fetch BOTTOM of PREG)))
              (TOP (SETQ VAL (fetch (REGION TOP) of (fetch REG of PANE)))
                   (if (IGREATERP VAL SCREENHEIGHT)
                       then (IDIFFERENCE (IDIFFERENCE (fetch TOP of PREG)
                                                      (IDIFFERENCE VAL SCREENHEIGHT))
                                   (WINDOWPROP PANE 'BORDER))
                     else (fetch TOP of PREG)))
              (LEFT (SETQ VAL (fetch (REGION LEFT) of (fetch REG of PANE)))
                    (if (ILESSP VAL 0)
                        then (IPLUS (WINDOWPROP PANE 'BORDER)
                                    (IDIFFERENCE (fetch LEFT of PREG)
                                           VAL))
                      else (fetch (REGION LEFT) of PREG)))
              (RIGHT (SETQ VAL (fetch (REGION RIGHT) of (fetch REG of PANE)))
                     (if (IGREATERP VAL SCREENWIDTH)
                         then (IDIFFERENCE (IDIFFERENCE (fetch RIGHT of PREG)
                                                        (IDIFFERENCE VAL SCREENWIDTH))
                                     (WINDOWPROP PANE 'BORDER))
                       else (fetch RIGHT of PREG)))
              (SHOULDNT])
```

(\TEDIT.PANE.SCREENREGION

```
  [LAMBDA (PANE)                                                  ; Edited 19-Nov-2023 23:58 by rmk

    ;; For scrolling when the window is partially offscreen.

    ;; If the clipping region is entirely onscreen, this returns the clipping region.

    ;; If it is off the screen at the bottom or top, this returns the subregion of the clipping region that is actually onscreen.

    ;; This assumes that the clipping region is in the window's coordinate system.

    (LET ((PREG (DSPCLIPPINGREGION NIL PANE))
          (YOFFSET (DSPYOFFSET NIL PANE))
          BOTTOM HEIGHT)
         (if (AND (IGEQ YOFFSET 0)
                  (ILEQ (IPLUS YOFFSET (fetch PTOP of PREG))
                        SCREENHEIGHT))
             then ;; Top and bottom are on the screen

                  PREG
           else (SETQ BOTTOM (IMAX 0 (IDIFFERENCE (fetch BOTTOM of PREG)
                                            YOFFSET)))
                (SETQ HEIGHT (IDIFFERENCE (IMIN (fetch PTOP of PREG)
                                                (IPLUS SCREENHEIGHT YOFFSET))
                                    BOTTOM))
                (create REGION using PREG BOTTOM _ BOTTOM HEIGHT _ HEIGHT])

)
```

;; 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 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 (FGETTOBJ TEXTOBJ TXTREADONLY)
        [bind (FIRSTTIME _ T) for CARET in (FGETTOBJ TEXTOBJ CARET) 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 21-Mar-2024 21:27 by rmk
                                                                              ; Edited 20-Mar-2024 06:46 by rmk
                                                                              ; Edited 15-Mar-2024 22:00 by rmk
                                                                              ; Edited  9-Mar-2024 23:39 by rmk
                                                                              ; Edited 21-Feb-2024 20:43 by rmk
                                                                              ; Edited 18-Feb-2024 23:35 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.
    (LET ((TEXTOBJ (TEXTOBJ TSTREAM)))
         (CL:UNLESS (FGETTOBJ TEXTOBJ TXTNEEDSUPDATE)
             (CL:UNLESS SEL
                 (SETQ SEL (\DTEST (FGETTOBJ TEXTOBJ SEL)
                                   'SELECTION)))
             (CL:WHEN (AND (FGETSEL SEL SET)
                           (IGREATERP (FGETTOBJ TEXTOBJ TEXTLEN)
                                   0))                                  ; If the selection isn't set, don't bother.

                 ;; This is essentially "find selpane in panes"  and the corresponding L1/LN in SEL.  SELPANE is the pane of the last selection
                 (for PANE CH# CARETYBOT TOPLINE PREG PHEIGHT PBOTTOM (SELPANE _ (OR (FGETTOBJ TEXTOBJ SELPANE)
                                                                                     (\TEDIT.PRIMARYW TEXTOBJ)))
                     inpanes TEXTOBJ as L1 inside (FGETSEL SEL L1) as LN inside (FGETSEL SEL LN)
                    first (CL:UNLESS SELPANE (RETURN)) when (EQ PANE SELPANE)
                    do ;; Find the YBOT in SELPANE of the line with the selected character, if known.
                       (SETQ PREG (DSPCLIPPINGREGION NIL PANE))
                       (SETQ PHEIGHT (fetch PTOP of PREG))
                       (SETQ PBOTTOM (fetch BOTTOM of PREG))

                       ;; Find the selected character either at the beginning or end of the selection.
                       [SETQ CH# (IMAX 1 (IMIN (TEXTLEN TEXTOBJ)
                                               (SELECTQ (FGETSEL SEL POINT)
                                                    (LEFT (FGETSEL SEL CH#))
                                                    (RIGHT (SUB1 (FGETSEL SEL CHLIM)))
                                                    (SHOULDNT]
                       (SETQ CARETYBOT (SELECTQ (FGETSEL SEL POINT)
                                           (LEFT (AND L1 (GETLD L1 YBOT)))
                                           (RIGHT (AND LN (GETLD LN YBOT)))
                                           NIL))

                       ;; We don't want to jump around if the caret is already visible in SELPANE, unless EVEN.IF.VISIBLE
                       (CL:WHEN (OR EVEN.IF.VISIBLE (NOT CARETYBOT)
                                    (IGEQ CARETYBOT PHEIGHT)
                                    (ILESSP CARETYBOT PBOTTOM))

                           ;; Not visible, work to do: Make sure the line containing CH# is linked in and scrolled to the top of the pane. This
                           ;; should be replaceable by \TEDIT.SCROLLFN, with the appropriate DY.
                           (\TEDIT.SHOWSEL SEL NIL SELPANE TEXTOBJ)
                           (SETQ TOPLINE (CADR (\TEDIT.LINES.ABOVE TEXTOBJ CH# PHEIGHT)))
                           (SETYPOS TOPLINE (IDIFFERENCE PHEIGHT (GETLD TOPLINE LHEIGHT)))
                           (LINKLD (fetch (TEXTWINDOW PLINES) of SELPANE)
                                  TOPLINE)

                           ;; Lines are established and positioned.  Clear the window, display the first (caret) caret line.
                           (\TEDIT.CLEARPANE PANE)
                           (\TEDIT.DISPLAYLINE TEXTOBJ TOPLINE SELPANE)
                           (\TEDIT.FILLPANE TOPLINE TEXTOBJ SELPANE)
                                                                        ; And fill out the window from there.
                           (\TEDIT.FIXSEL SEL TEXTOBJ NIL SELPANE)
                           (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ SELPANE)
                           (\TEDIT.SHOWSEL SEL T SELPANE TEXTOBJ))
                       (RETURN))))])
```

(\TEDIT.SETCARET

```
  [LAMBDA (SEL PANE TEXTOBJ DISPOSITION CARET)                          ; Edited 15-Dec-2023 23:37 by rmk
                                                                        ; Edited 17-Nov-2023 23:55 by rmk
                                                                        ; Edited 26-Oct-2023 08:42 by rmk
                                                                        ; Edited 24-Oct-2023 11:50 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 is optional.  If not given, then PANE's caret in TEXTOBJ is used. 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.
    (CL:WHEN (FGETSEL SEL HASCARET)
        (CL:UNLESS CARET
            (SETQ CARET (for P inpanes (PROGN TEXTOBJ) as PCARET in (FGETTOBJ TEXTOBJ CARET)
                           when (EQ P PANE) do (RETURN PCARET))))
        [SELECTQ DISPOSITION
             (DISABLE (\TEDIT.UPCARET CARET -10 -10))
             ((NIL OFF)                                                 ; Originally, no X Y, like DISABLE. But that left turds
                  (\TEDIT.UPCARET CARET -10 -10))
             (LET (LINE X Y PREG)                                       ; 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 (EQ P PANE) do (SELECTQ (FGETSEL SEL POINT)
                                             (LEFT (CL:WHEN (AND L1 (WITHINLINEP (FGETSEL SEL CH#)
                                                                             L1))
                                                        (SETQ LINE L1)
                                                        (SETQ X (FGETSEL SEL X0))))
                                             (RIGHT (CL:WHEN (AND LN (WITHINLINEP (SUB1 (FGETSEL SEL CHLIM))
                                                                                LN))
                                                                  (SETQ LINE LN)
                                                                  (SETQ X (FGETSEL SEL XLIM))))
                                                       (SHOULDNT))
                                                  (RETURN))
                         (if LINE
                             then (SETQ Y (FGETLD LINE YBASE))
                                  (SETQ PREG (DSPCLIPPINGREGION NIL PANE))
                                  (COND
                                     ((AND (ILESSP Y (fetch PTOP of PREG))
                                           (IGEQ (FGETLD LINE YBOT)
                                                 (fetch BOTTOM of PREG)))   ; Move to right position even if not flashing
                                      (CL:IF (FGETTOBJ TEXTOBJ TXTREADONLY)
                                          (\TEDIT.UPCARET CARET X Y)
                                          (\TEDIT.DOWNCARET CARET X Y)))
                                     (T ;; 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 (CARETS)                                                         (* jds "12-Jul-85 11:18")
                                                                           (* Reset the caret to its normal state state, from the selection
                                                                           caret)
    (for CARET inside CARETS do (replace TCCARET of CARET with (\CARET.CREATE BXCARET])
)
```

;; Menu interfacing

```
(DEFINEQ
```

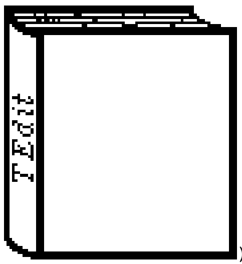
(**TEDIT.ADD.MENUITEM**

```
  [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 (W)                                                              ; Edited 20-Mar-2024 11:02 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 ((TEXTOBJ (fetch (TEXTWINDOW WTEXTOBJ) of W))
           (WMENU (WINDOWPROP W 'TEDIT.MENU))
           THISMENU CH OFILE OCURSOR LINES SEL ITEM)
          (TEXTOBJ! TEXTOBJ)
          (COND
             ((EQ (FGETTOBJ TEXTOBJ EDITOPACTIVE)
                  T)

              ;; We're busy doing something, but not sure what.  Give a general 'please wait' msg:

              (TEDIT.PROMPTPRINT TEXTOBJ "Edit operation in progress; please wait." T)
              (RETURN))
             ((FGETTOBJ TEXTOBJ EDITOPACTIVE)
              ;; We know specifically what's happening.  Tell him:
              (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (FGETTOBJ TEXTOBJ EDITOPACTIVE)
                                                " in progress; please wait.")
                     T)
              (RETURN)))
          (SETQ THISMENU (COND
                            (WMENU)
                            ((SETQ WMENU (WINDOWPROP W 'TEDIT.MENU.COMMANDS))
                             (PROG1 (SETQ WMENU (\TEDIT.CREATEMENU WMENU))
                                 (WINDOWPROP W 'TEDIT.MENU WMENU)))
                            (TEDIT.DEFAULT.MENU)))
          (SETQ ITEM (MENU THISMENU))
          (ERSETQ (RESETLST
                     [RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ)
                             '(AND (\TEDIT.MARKINACTIVE OLDVALUE]
                     (SETTOBJ TEXTOBJ EDITOPACTIVE (OR (CAR ITEM)
                                                       T))       ; So we ca ntell the guy WHAT op is active.
                     [SELECTQ (CAR 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                                   ; Stop this session.
                                 (\TEDIT.QUIT W))
                         (Substitute                             ; Search-and-replace
                                      (RESETLST
                                          (RESETSAVE (CURSOR WAITINGCURSOR))
                                          (TEDIT.SUBSTITUTE TEXTOBJ)))
                         (Find                                   ; Case sensitive search, with * and # wildcards
                               [SETQ OFILE (TEDIT.GETINPUT TEXTOBJ "Text to find: " (\TEDIT.GET.TARGET.STRING
                                                                                      TEXTOBJ
                                                                                      'TEDIT.LAST.FIND.STRING]
                              (CL:WHEN OFILE
                                   (SETQ SEL (TEXTSEL TEXTOBJ))
                                   (\TEDIT.SHOWSEL SEL NIL)
                                   (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)
                                   (SETQ CH (TEDIT.FIND TEXTOBJ (MKSTRING OFILE)
                                                   NIL NIL T))
                                   (COND
                                      (CH                        ; We found the target text.
                                          (TEDIT.PROMPTPRINT TEXTOBJ "Done.")
                                          (SETSEL SEL CH# (CAR CH))
                                                                 ; Set up SELECTION to be the found text
                                          (SETSEL SEL CHLIM (ADD1 (CADR CH)))
                                          [SETSEL SEL DCH (ADD1 (IDIFFERENCE (CADR CH)
                                                                        (CAR CH]
                                          (SETSEL SEL POINT 'RIGHT)
                                          (SETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL
                                                                             ))
                                          (\TEDIT.RESET.EXTEND.PENDING.DELETE SEL TEXTOBJ)
                                                                 ; And never pending a deletion.
                                          (\TEDIT.FIXSEL SEL TEXTOBJ)
                                          (TEDIT.NORMALIZECARET TEXTOBJ)
                                          (\TEDIT.SHOWSEL SEL T)
                                                                 ; And get it into the TEXTOBJ
                                          (PUTTEXTPROP TEXTOBJ 'TEDIT.LAST.FIND.STRING OFILE))
                                      (T (TEDIT.PROMPTPRINT TEXTOBJ "(not found)")
                                         (\TEDIT.SHOWSEL SEL T)))))
                         (Looks                                  ; He wants to set the font for the current selection
                                 (\TEDIT.LOOKS TEXTOBJ))
                         (Hardcopy                               ; Print this document
                                   (TEDIT.HARDCOPY TEXTOBJ))
                         (Press% File                            ; Make a hardcopy file with this document in it.
                                      (TEDIT.HCPYFILE TEXTOBJ))
                         (Expanded% Menu                         ; Open the expanded operations menu.
                              (\TEDIT.EXPANDED.MENU TEXTOBJ))
                         (Character% Looks                       ; Open the menu for setting character looks
                              (\TEDIT.EXPANDEDCHARLOOKS.MENU TEXTOBJ))
                         (Paragraph% Formatting                  ; Open the paragraph formatting menu
                              (\TEDIT.EXPANDEDPARA.MENU TEXTOBJ))
                         (Page% Layout                           ; Open the page-layout menu
                                       (\TEDIT.MENU.START (COPYTEXTSTREAM TEDIT.EXPANDED.PAGEMENU T)
                                              (\TEDIT.PRIMARYW TEXTOBJ)
                                              "Page Layout Menu" 150 'PAGE))
                         (CL:WHEN (CAR ITEM)                     ; Apply a user-supplied function to the text stream
                             (APPLY* (CAR ITEM)
                                    (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)))])])
```

```
(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 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 _ '\TEDIT.MENU.WHENHELDFN
           WHENSELECTEDFN _ '\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