

File created: 4-May-95 10:37:05 {DSK}<lispcore>library>new>TEDIT.;5

changes to: (FNS \TEDIT.INSERT.PIECES TEDIT.COPY TEDIT.MOVE)
(FILES TEDITCOMMAND TEDITFILE TEDITFNKEYS TEDITHISTORY TEDITLOOKS TEDITPAGE TEDITWINDOW)

previous date: 22-Mar-95 18:17:12 {DSK}<lispcore>library>new>TEDIT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1995 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **TEDITCOMS**

```
( (FILES TEDITDECLS)
  (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (\SCRATCHLEN 64))
    (FILES (LOADCOMP)
      TEDITDECLS))
  (FILES PCTREE TEXTOFD TEDITCOMMAND TEDITSCREEN TEDITABBREV TEDITLOOKS)
  (VARS (TEDIT.TERMSA.FONTS NIL)
    (TEDIT.TENTATIVE NIL)
    (TEDIT.DEFAULT.PROPS NIL)
    (TEDIT.STARTUP.MONITORLOCK (CREATE.MONITORLOCK 'TEDIT.STARTUP))
    (TEDIT.RESTART.MENU (\CREATE.TEDIT.RESTART.MENU))
    ; Original was (CREATE MENU ITEMS_ '(NewEditProcess)).
    ; Changed by yabu.fx, for SUNLOADUP without DWIM.

  )
  (GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)
  (FNS \TEDIT2 COERCETEXTOBJ TEDIT TEDIT.CHARWIDTH TEDIT.COPY TEDIT.DELETE TEDIT.DO.BLUEPENDINGDELETE
    TEDIT.INSERT TEDIT.KILL TEDIT.MAPLINES TEDIT.MAPPICES TEDIT.MOVE TEDIT.QUIT TEDIT.STRINGWIDTH
    TEDIT.\INSERT TEXTOBJ TEXTSTREAM \TEDIT.INCLUDE \TEDIT.INSERT.PIECES \TEDIT.MOVE.PIECEMAPFN
    \TEDIT.OBJECT.SHOWSEL \TEDIT.RESTARTFN \TEDIT.CHARDELETE \TEDIT.COPY.PIECEMAPFN \TEDIT.DELETE
    \TEDIT.DIFFUSE.PARALOOKS \TEDIT.FOREIGN.COPY? \TEDIT.QUIT \TEDIT.WORDDELETE \TEDIT1)
  (P (MOVD? 'NIL 'OBJECTOUTOFTEDIT))
    ; HOOK for looked-string copy, etc. Used in
    ; \TEDIT.FOREIGN.COPY?.

  (COMS (FNS \CREATE.TEDIT.RESTART.MENU))
    ; Added by yabu.fx, for SUNLOADUP without DWIM.
    ; Debugging functions

  (COMS
    (FNS PLCHAIN PRINTLINE SEEFIL))
    ; Object-oriented editing

  (COMS
    (FNS TEDIT.INSERT.OBJECT TEDIT.EDIT.OBJECT TEDIT.FIND.OBJECT TEDIT.FIND.OBJECT.SUBTREE
      TEDIT.PUT.OBJECT TEDIT.GET.OBJECT TEDIT.OBJECT.CHANGED))
  (FILES TEDITFIND TEDITHISTORY TEDITFILE TEDITWINDOW TEDITSELECTION IMAGEOBJ TFBRAVO TEDITHCPY TEDITPAGE
    TEDITMENU TEDITFNKEYS)
    ; TEDIT Support information

  (COMS
    (E (SETQ TEDITSYSTEMDATE (DATE)))
    (VARS TEDITSYSTEMDATE (TEDITSYSTEMDATE "TeditSupport.PA"))
    (FNS MAKETEDITFORM)
    (P (ADDTTOVAR LAFITESPECIALFORMS ("Tedit Report" 'MAKETEDITFORM "Report a problem with Tedit"))
      (SETQ LAFITEFORMSMENU NIL)))
  (COMS
    ; LISTFILES Interface, so the system can decide if a file is a
    ; Tedit file.

    (ADDVARS (PRINTFILETYPES (TEDIT (TEST \TEDIT.FORMATTEDP1)
      (EXTENSION (TEDIT]))
```

(FILESLOAD TEDITDECLS)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ \SCRATCHLEN 64)

(CONSTANTS (\SCRATCHLEN 64))

)

(FILESLOAD (LOADCOMP)
 TEDITDECLS)

)

(FILESLOAD PCTREE TEXTOFD TEDITCOMMAND TEDITSCREEN TEDITABBREV TEDITLOOKS)

(RPAQQ **TEDIT.TERMSA.FONTS** NIL)

(RPAQQ **TEDIT.TENTATIVE** NIL)

(RPAQQ **TEDIT.DEFAULT.PROPS** NIL)

(RPAQ **TEDIT.STARTUP.MONITORLOCK** (CREATE.MONITORLOCK 'TEDIT.STARTUP))

(RPAQ **TEDIT.RESTART.MENU** (\CREATE.TEDIT.RESTART.MENU))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)
)

(DEFINEQ

(\TEDIT2

[LAMBDA (TEXT WINDOW UNSPAWNED)

; Edited 12-Jun-90 17:51 by mitani

;; Does the actual editing work, once TEDIT has OPENTEXTSTREAMed the thing to be edited.

```

(\TEDIT.COMMAND.LOOP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)) ; Run the editing engine
(CLOSEW WINDOW) ; Close the edit window
(\TEXTCLOSEF TEXT) ; Close the underlying files
(replace (STREAM ACCESSBITS) of TEXT with BothBits) ; But leave the stream itself accessible
(AND (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
  'AFTERQUITFN)
  (APPLY* (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
    'AFTERQUITFN)
    WINDOW TEXT)) ; Apply any post-window-close (and post-QUIT) function

```

```

(COND
  (UNSPAWNED ; We're not a distinct process: Send back the edited text in some
    ; suitable form

```

```

  (COND
    ((NEQ (fetch (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT))
      T)
      (PROG1 (fetch (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT))
        (replace (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT) with NIL)))
    ((STRINGP (fetch (TEXTOBJ TXTFILE) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT)))
      (COERCETEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
        'STRINGP))
    (T TEXT))

```

(COERCETEXTOBJ

[LAMBDA (STREAM TYPE OUTPUTSTREAM)

; Edited 18-Apr-93 23:42 by jds

;; Coerce the contents of the TEXOBJ to be of the given type. This is for making a string from a textobj, e.g.

```

(PROG ((TEXTOBJ (COND
  ((type? STREAM STREAM)
    (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
  (T STREAM)))
  OFILE FMTFILE)

```

```

(OR (type? TEXTOBJ TEXTOBJ)
  (\ILLEGAL.ARG TEXTOBJ))

```

; If we haven't got a TEXTOBJ, something is wrong.

```

(RETURN (SELECTQ TYPE
  ((STRINGP STRING)

```

```

    (AND (ILEQ (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
      65535)

```

```

    (PROG ((STR (ALLOCSTRING (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))
      PC

```

```

      (CH# 1)
      (PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
      (DELTA 0)
      PFILE)
      (SETQ PC (\GETBASEPTR (\FIRSTNODE PCTB)
        0))

```

[WHILE PC do (COND

```

  ((ATOM PC) ; It's the lastpiece atom -- do nothing.
    (SETQ PC NIL))
  ((fetch CLINVISIBLE of (fetch (PIECE PLOOKS) of PC))

```

```

;; If the characters are invisible, do nothing. HOWEVER, we have to shrink the final string to
;; account for the characters we ignored.

```

```

  (add DELTA (fetch (PIECE PLEN) of PC)))
  ((fetch (PIECE PSTR) of PC)
    [OR (ZEROP (fetch (PIECE PLEN) of PC))
      (RPLSTRING STR CH# (SUBSTRING (fetch (PIECE PSTR)
        of PC)

```

```

        1
        (fetch (PIECE PLEN)
          of PC]

```

```

  (add CH# (fetch (PIECE PLEN) of PC)))
  ((SETQ PFILE (fetch (PIECE PFILE) of PC))
    [COND

```

```

      ((NOT (OPENP PFILE))
        (SETQ PFILE (\TEDIT.REOPEN.STREAM STREAM PFILE))
        (SETFILEPTR PFILE (fetch (PIECE PPOS) of PC))
        (for C from CH# as I from 1
          to (fetch (PIECE PLEN) of PC)
            do (RPLCHARCODE STR C (BIN PFILE)))
        (add CH# (fetch (PIECE PLEN) of PC)))
      ((fetch (PIECE POBJ) of PC)
        ; DO NOTHING FOR OBJECTS
        (add CH# (fetch (PIECE PLEN) of PC))
        (add DELTA (fetch (PIECE PLEN) of PC)))
      (T (ERROR "CANNOT GET TEXT FROM A 'PIECE.'" PC)))

```

```

                                (AND PC (SETQ PC (FETCH (PIECE NEXTPIECE) OF PC])
                                [COND
                                  ((ZEROP DELTA) ; No change in the length; do nothing.
                                   )
                                   (T ; The string got shortened to account for invisible chars. Chop it
                                      ; off
                                      (SETQ STR (SUBSTRING STR 1 (IDIFFERENCE (fetch (TEXTOBJ TEXTLEN)
                                                                                   of TEXTOBJ)
                                                                                   DELTA])
                                      (RETURN STR))))
                                (STREAM (COND
                                  ((type? STREAM (fetch (TEXTOBJ TXTFILE) of TEXTOBJ))
                                   (OPENFILE (fetch (STREAM FULLNAME) of (fetch (TEXTOBJ TXTFILE) of TEXTOBJ))
                                    'INPUT)
                                   (replace (STREAM ACCESSBITS) of (fetch (TEXTOBJ TXTFILE) of TEXTOBJ)
                                    with ReadBit)))
                                  (\SETUPGETCH 1 TEXTOBJ)
                                  (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ))
                                  (FILE [SETQ OFILE (OR (AND OUTPUTSTREAM (OPENP OUTPUTSTREAM 'OUTPUT))
                                                         (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW]
                                                         (TEDIT.PUT.PCTB TEXTOBJ OFILE)
                                                         (OR OUTPUTSTREAM (CLOSEF OFILE))
                                                         OFILE)
                                  (SPLIT ; i.e., Return 2 files, one with plain text, one with formatting info, such that concatenating them will do the right
                                      ; thing.
                                      (SETQ OFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW))
                                      (SETQ FMTFILE (CAR (TEDIT.PUT.PCTB TEXTOBJ (\GETSTREAM OFILE 'BOTH)
                                                                                   NIL T)))
                                      (CLOSEF OFILE)
                                      (CONS OFILE FMTFILE))
                                  NIL))

```

(TEDIT

```

[LAMBDA (TEXT WINDOW DONTSPAWN PROPS) ; Edited 3-Jun-88 14:27 by jds
;; User entry to the text editor. Takes an optional window to be used for editing
;; DONTSPAWN => Don't try to create a new process for this edit.
(PROG (PROC TEDITCREATEDWINDOW) ; Include the default properties in the list.
[COND
  ((AND TEXT (ATOM TEXT)) ; Make sure the file exists before trying to open the window.
   (SETQ TEXT (OPENFILE TEXT 'INPUT 'OLD])
  (RESETLST
   [RESETSAVE NIL `(AND ,WINDOW (WINDOWPROP ,WINDOW 'TEXTOBJ NIL]
   (WITH.MONITOR TEDIT.STARTUP.MONITORLOCK
    (COND
      ((NOT WINDOW)
       (SETQ TEDITCREATEDWINDOW T)
       (SETQ WINDOW (COND
         [(OR (NOT TEDIT.DEFAULT.WINDOW)
              (\TEDIT.ACTIVE.WINDOWP TEDIT.DEFAULT.WINDOW))
          (TEDIT.CREATEW (COND
                        ((AND TEXT (ATOM TEXT))
                         (CONCAT "Please specify an editing window for " TEXT))
                        (T "Please specify a region for the editing window."))
          TEXT
          (APPEND PROPS (COPY TEDIT.DEFAULT.PROPS]
          (T (\TEDIT.CREATEW.FROM.REGION (WINDOWPROP TEDIT.DEFAULT.WINDOW
                                                    'REGION)
          TEXT
          (APPEND PROPS (COPY TEDIT.DEFAULT.PROPS)))
          ; Replace the old title
          TEDIT.DEFAULT.WINDOW)))
      (WINDOWPROP WINDOW 'TEXTOBJ T) ; Mark the newly-created window reserved until the
                                      ; OPENTEXTSTREAM has done its work.
    ))
    ; mark that we created the window so that we know we can
    ; update the title, etc.
  ))
)
[SETQ TEXT (OPENTEXTSTREAM TEXT WINDOW NIL (APPEND PROPS '(BEING-EDITED T)
; Connect the editor to the window
(replace (TEXTOBJ TXTEDITING) of (TEXTOBJ TEXT) with T) ; For the moment, mark the document as actively in edit. (so
; caret flashes when the window is first brought up.)
[COND
  (TEDITCREATEDWINDOW (TEXTPROP TEXT 'TEDITCREATEDWINDOW 'T)
  (COND
    (DONTSPAWN ; Either no processes running, or specifically not to spawn one.
     (RETURN (\TEDIT2 TEXT WINDOW T)))
    (T ; Spawn a process to do the edit.
     [SETQ PROC (ADD.PROCESS (LIST '\TEDIT2 (KWOTE TEXT)
                                   WINDOW NIL)
                           'NAME
                           'TEdit
                           'RESTARTABLE

```

```

        'HARDRESET
        'RESTARTFORM
        (LIST 'TEDIT.RESTARTFN (KWOTE TEXT)
              WINDOW
              (KWOTE PROPS])
        (PROCESSPROP PROC 'WINDOW WINDOW)
        (COND
          ((NOT (LISTGET (APPEND PROPS (COPY TEDIT.DEFAULT.PROPS))
                        'LEAVETTY))
           ; Unless he asked us to leave the tty where it is, TEdit should get
           ; it.
           (TTY.PROCESS PROC)))
        (RETURN PROC])

```

(TEDIT.CHARWIDTH

[LAMBDA (CH FONT TERMSA)

(* jds "22-OCT-83 19:32")

(* Returns the width of CH in FONT printed according to any special printing instructions in CHARTABLE TERMSA)

```

(COND
  (TERMSA
    (SELECTC (fetch CCECHO of (\SYNCODE TERMSA CH))
      (INDICATE.CCE (IPLUS (COND
        ((IGREATERP CH 127) (* META character)
         (SETQ CH (LOGAND CH 127))
         (CHARWIDTH (CHARCODE %#)
                     FONT))
        (T 0))
        (COND
          ((ILESSP CH 32) (* CONTROL character)
           (SETQ CH (LOGOR CH 64))
           (CHARWIDTH (CHARCODE ^)
                       FONT))
          (T 0))
        (CHARWIDTH CH FONT)))
      (SIMULATE.CCE (SELCHARQ CH
        ((EOL CR LF)
         (IMAX 6 (CHARWIDTH CH FONT)))
        (ESCAPE (CHARWIDTH (CHARCODE $)
                           FONT))
        (BELL 0)
        (TAB 36)
        (CHARWIDTH CH FONT)))
      (REAL.CCE (CHARWIDTH CH FONT))
      (IGNORE.CCE 0)
      (SHOULDNT)))
    (T
      (SELCHARQ CH
        (CR (IMAX 6 (CHARWIDTH CH FONT)))
        (TAB 36)
        (CHARWIDTH CH FONT)))
      (* The usual case is to treat every character as a graphic.)

```

(TEDIT.COPY

[LAMBDA (FROM TO)

; Edited 2-May-95 12:23 by sybalsky:mv:envos

; First, Turn off the global flag that got us here.

```

  (SETQ TEDIT.COPY.PENDING NIL)
  (COND
    ((NOT (AND FROM (fetch (SELECTION SET) of FROM)))
     ; There MUST be a source selected first.
     (TEDIT.PROMPTPRINT (fetch (SELECTION \TEXTOBJ) of TO)
      "Copy source selection hasn't been set yet." T))
    ((ZEROP (fetch (SELECTION DCH) of FROM))
     ; The source is empty. Just turn off the selection hilite and
     ; ignore the request.
     (\SHOWSEL FROM NIL NIL))
    ((fetch (TEXTOBJ TXTREADONLY) of (fetch (SELECTION \TEXTOBJ) of TO))
     ; The target is read-only. Don't do anything except turn off the
     ; selection highlighting and ignore the request.
     (\SHOWSEL FROM NIL NIL))
    (T (\SHOWSEL FROM NIL NIL)
     ; Before all else, make sure the copy source selection is turned
     ; off
     (replace (SELECTION SET) of FROM with NIL)
     (COND
       ((AND TO (fetch (SELECTION SET) of TO))
        ; Can only do copy if there's a target selection
        (PROG ((TOOBJ (fetch (SELECTION \TEXTOBJ) of TO))
          (FROMOBJ (fetch (SELECTION \TEXTOBJ) of FROM))
          (CROSSCOPY (NEQ (fetch (SELECTION \TEXTOBJ) of FROM)
                        (fetch (SELECTION \TEXTOBJ) of TO)))
          TOLEN LEN INSPC INSPC# PC NPC PCCH NPCCH OPLEN EVENT REPLACING INSERTCH# PCLST OBJ COPYFN
          UNDOCHAIN)
          (SETQ PCLST (TEDIT.SELECTED.PIECES FROMOBJ FROM CROSSCOPY (FUNCTION \TEDIT.COPY.PIECEMAPFN)
                                FROMOBJ TOOBJ))
          ; Get the list of pieces to be copied
          (SETQ REPLACING (TEDIT.DO.BLUEPENDINGDELETE TO TOOBJ))
          ; Do any blue-pending-delete
          (SETQ TOLEN (fetch (TEXTOBJ TEXTLEN) of TOOBJ))
          (\SHOWSEL TO NIL NIL)
          ; NOW turn off the target selection.
          [COND
            ((EQ (fetch (SELECTION POINT) of TO)
              'LEFT)

```

```

      (SETQ INSERTCH# (fetch (SELECTION CH#) of TO)))
      (T (SETQ INSERTCH# (IMIN (fetch (SELECTION CHLIM) of TO)
                                (ADD1 TOLEN) ; Figure out where to do the insertion.

(COND
  ((AND (fetch (TEXTOBJ FORMATTEDP) of FROMOBJ)
        (NOT (fetch (TEXTOBJ FORMATTEDP) of TOOBJ)))
    ; The source is formatted and the target isn't. Give the guy a
    ; choice.
    ; For now, convert the target file to formatted.
    (\TEDIT.CONVERT.TO.FORMATTED TOOBJ)))
  (SETQ UNDOCHAIN (\TEDIT.INSERT.PIECES TOOBJ INSERTCH# PCLST (SETQ LEN
                                                                (IDIFFERENCE (fetch (SELECTION
                                                                CHLIM)
                                                                of FROM)
                                                                (fetch (SELECTION CH#)
                                                                of FROM))))
    NIL NIL CROSSCOPY NIL T))
  (bind OBJ AFTERCOPYFN for PC in PCLST when [AND (SETQ OBJ (fetch (PIECE POBJ) of PC))
                                                  (SETQ AFTERCOPYFN (IMAGEOBJPROP OBJ
                                                                'AFTERCOPYFN))
    do (APPLY* AFTERCOPYFN OBJ))
  (SETQ EVENT (fetch (TEXTOBJ TXTHISTORY) of TOOBJ))
  (\TEDIT.HISTORYADD TOOBJ (create TEDITHISTORYEVENT
                                   THACTION _ (COND
                                                (REPLACING 'Replace)
                                                (T 'Copy))
                                   THLEN _ LEN
                                   THCH# _ INSERTCH#
                                   THFIRSTPIECE _ (LIST UNDOCHAIN)
                                   THOLDINFO _ (AND REPLACING EVENT)))
    ; Make a history-list entry for the COPY.
  (replace (TEXTOBJ \DIRTY) of TOOBJ with T) ; Mark the document changed
  (replace (TEXTOBJ TEXTLEN) of TOOBJ with (IPLUS LEN TOLEN))
    ; Set the new length
  (\FIXILINES TOOBJ TO INSERTCH# LEN TOLEN) ; Fix LINES and SELs
  [COND
    ((EQ (fetch (TEXTOBJ FORMATTEDP) of TOOBJ)
         (fetch (TEXTOBJ FORMATTEDP) of FROMOBJ))
      ; Either both of the files are formatted or neither is. This case is
      ; OK
    )
    ((fetch (TEXTOBJ FORMATTEDP) of TOOBJ)
      ; The source wasn't formatted, but the target is. Go convert the
      ; copied text.
      (\TEDIT.CONVERT.TO.FORMATTED TOOBJ INSERTCH# (IPLUS INSERTCH# LEN)
      (TEDIT.UPDATE.SCREEN TOOBJ) ; Refresh the display
      (replace (SELECTION CH#) of TO with INSERTCH#) ; Correct the target selection
      (replace (SELECTION CHLIM) of TO with (IPLUS INSERTCH# LEN))
      (replace (SELECTION DCH) of TO with LEN)
      (replace (SELECTION DX) of TO with 0)
      (replace (SELECTION POINT) of TO with 'RIGHT) ; (replace CARETLOOKS of TOOBJ with
      ; (TEDIT.GET.INSERT.CHARLOOKS TOOBJ TO))
      ; Make any later type-in look like what we just copied.
      (replace (TEXTOBJ \INSERTPCVALID) of TOOBJ with NIL)
      ; And make sure that the pieces copied never have their strings
      ; smashed by back spacing.
      (replace (TEXTOBJ \INSERTPCVALID) of FROMOBJ with NIL)
      (\FIXSEL TO TOOBJ)
      (\SHOWSEL TO NIL T)))
  (T
    ; There is no target selection -- complain
    (TEDIT.PROMPTPRINT (fetch (SELECTION \TEXTOBJ) of FROM)
      "Please select a destination for the copy first." T])

```

(TEDIT.DELETE

[LAMBDA (STREAM SEL LEN LEAVECARETLOOKS)

; Edited 12-Jun-90 17:49 by mitani

;; DELETE THE CHARACTERS SPECIFIED FROM THE MAIN TEXT.

;; If LEAVECARETLOOKS is non-NIL, the selection will NOT be set up to do the right thing with type-in. This can save time in inner loops.

```

(PROG ((TEXTOBJ (TEXTOBJ STREAM)))
  [COND
    ((FIXP SEL)
      (TEDIT.SETSEL STREAM SEL LEN NIL NIL LEAVECARETLOOKS)
      (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
      (OR SEL (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)))
      (\TEDIT.DELETE SEL TEXTOBJ))

```

(TEDIT.DO.BLUEPENDINGDELETE

[LAMBDA (SEL TEXTOBJ)

; Edited 29-May-91 18:21 by jds

(* Check for blue-pending-delete, and do it if it's there.)

(* Return T if the deletion was made.

For people who need to know)

```

(COND
  ((fetch (TEXTOBJ BLUEPENDINGDELETE) of TEXTOBJ)
    (PROG1 (fetch (TEXTOBJ BLUEPENDINGDELETE) of TEXTOBJ)
      (COND

```

(* If he's in a Blue-pending-delete state, delete the selection.)

```

((NOT (ZEROP (fetch (SELECTION DCH) of SEL))) (* There really IS something to delete.)
(\SHOWSEL SEL NIL NIL) (* Turn off the selection)
(\DELETECH (fetch (SELECTION CH#) of SEL)
(fetch (SELECTION CHLIM) of SEL)
(fetch (SELECTION DCH) of SEL)
TEXTOBJ) (* Delete the characters.)
(\FIXDLINES (fetch (TEXTOBJ LINES) of TEXTOBJ)
SEL
(fetch (SELECTION CH#) of SEL)
(fetch (SELECTION CHLIM) of SEL)
TEXTOBJ) (* Fix up any line descriptors to reflect the deletion.)
(TEDIT.RESET.EXTEND.PENDING.DELETE SEL) (* Make it a normal selection again.)
(replace (SELECTION CHLIM) of SEL with (fetch (SELECTION CH#) of SEL))

(* Fix up the selection, so that it is 0 wide, where the old text used to be.)

(replace (SELECTION DCH) of SEL with 0)
(replace (SELECTION POINT) of SEL with 'LEFT)
(\FIXSEL SEL TEXTOBJ) (* Make its line descriptors &c reflect the new reality)
(\SHOWSEL SEL NIL T) (* And turn it back on.)
)
(T (* Don't do it, since it's zero-width.
However, DO turn off the blue-pendingness of it.)

(TEDIT.RESET.EXTEND.PENDING.DELETE SEL))))))

```

(TEDIT.INSERT

```

[LAMBDA (STREAM TEXT CH#ORSEL LOOKS DONTSCROLL) ; Edited 29-May-91 18:21 by jds
; Insert TEXT (character, litatom or string) at the appropriate spot
; in the text.

(SETQ STREAM (TEXTSTREAM STREAM))
[COND
  ((FIXP CH#ORSEL) ; He gave us a ch# to insert before
  (TEDIT.SETSEL STREAM CH#ORSEL 1 'LEFT)
[COND
  ((LITATOM TEXT)
  (SETQ TEXT (MKSTRING TEXT))
[OR (type? SELECTION CH#ORSEL)
  (SETQ CH#ORSEL (fetch (TEXTOBJ SEL) of (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
[COND
  ((AND (STRINGP TEXT)
  (ZEROP (NCHARS TEXT))) ; Can't insert an empty string sensibly. It confuses the screen
; update code.

NIL)
[ (AND CH#ORSEL (fetch (SELECTION SET) of CH#ORSEL))
  (PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
  CH# LINE XPOINT OTEXTLEN DS LINES CHARS BLANKSEEN CRSEEN)
  (TEDIT.DO.BLUEPENDINGDELETE CH#ORSEL TEXTOBJ) ; If the selected text was for pending delete, delete it before
; doing the insert.

(COND
  (LOOKS ; If looks for this insertion were specified, set them up.
  (TEDIT.CARETLOOKS STREAM LOOKS)))
(SETQ OTEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)) ; The PRE-INSERT text length, for starting the screen update
; process

(COND
  ((fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
  ;; If this text is in a window, move it so the insertion point is on-screen, then turn off the selection highlight
  (COND
    ((NOT DONTSCROLL) ; If DONTSCROLL is T, then don't bother scrolling the window to
; show the change.
    (TEDIT.NORMALIZECARET TEXTOBJ CH#ORSEL)))
  (\SHOWSEL CH#ORSEL NIL NIL)))
(SETQ CH# (IMAX 1 (SELECTQ (fetch (SELECTION POINT) of CH#ORSEL)
  (LEFT (fetch (SELECTION CH#) of CH#ORSEL))
  (RIGHT (IMIN (fetch (SELECTION CHLIM) of CH#ORSEL)
  (ADD1 (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))))
  NIL)))
(SETQ XPOINT (SELECTQ (fetch (SELECTION POINT) of CH#ORSEL)
  (LEFT (fetch (SELECTION X0) of CH#ORSEL))
  (RIGHT (fetch (SELECTION XLIM) of CH#ORSEL))
  NIL))
[COND
  [(type? STRINGP TEXT) ; It's a string: Count the characters and Insert them one by one
; into the text stream
  (SETQ CHARS (NCHARS TEXT))
  (for ACHAR instring TEXT as NCH# from CH# by 1 do (SELCHARQ ACHAR
    ((CR %#^M 1, CR)
    (SETQ CRSEEN T)
    (\INSERTCR ACHAR NCH# TEXTOBJ))
    (SPACE (SETQ BLANKSEEN T)
    (\INSERTCH ACHAR NCH# TEXTOBJ))
    (\INSERTCH ACHAR NCH# TEXTOBJ]
  (T ; It's a single character. Just insert it.
  (SETQ CHARS 1)
  (SELCHARQ TEXT

```

```

        ((CR %#^M 1,CR)
         (SETQ CRSEEN T)
         (\INSERTCR TEXT CH# TEXTOBJ))
        (SPACE (SETQ BLANKSEEN T)
         (\INSERTCH TEXT CH# TEXTOBJ))
        (\INSERTCH TEXT CH# TEXTOBJ)
        (\FIXILINES TEXTOBJ CH#ORSEL CH# CHARS OTEXTLEN) ; Fix up the line descriptors and the Selection.
        (COND
         ((NOT (fetch (TEXTOBJ TXTREADONLY) of (TEXTOBJ STREAM)))
          ; Update the edit window.
          (TEDIT.INSERT.UPDATESCREEN TEXT CH# CHARS XPOINT TEXTOBJ CH#ORSEL OTEXTLEN BLANKSEEN CRSEEN
           DONTSCROLL)
          ((NOT (fetch (TEXTOBJ TXTREADONLY) of (TEXTOBJ STREAM)))
           (TEDIT.PROMPTPRINT (TEXTOBJ STREAM)
            "Please select a place for the insertion." T))

```

(TEDIT.KILL

```

[LAMBDA (STREAM)
; Edited 12-Jun-90 17:49 by mitani
; (* Force the edit session supported by STREAM to terminate,
; and to return VALUE)

(COND
 ((type? STREAM STREAM)
  (SETQ STREAM (fetch (TEXTSTREAM TEXTOBJ) of STREAM)))
 (* If he gave us a textofd, get the textobj)
 [ (WINDOWP STREAM)
  (SETQ STREAM (WINDOWPROP STREAM 'TEXTOBJ))
  (* Take a window, and do the obvious with it.)
  ((type? TEXTOBJ STREAM)
   (* A Textobj is just fine)
   )
  (* Anything else is ungood, double-plus)
  (T
   (\ILLEGAL.ARG STREAM)))
 (replace (TEXTOBJ EDITFINISHEDFLG) of STREAM with T)
 (PROG (TEDW TEDPROC)
  (AND (SETQ TEDW (CAR (fetch (TEXTOBJ \WINDOW) of STREAM)))
   [PROCESSP (SETQ TEDPROC (WINDOWPROP TEDW 'PROCESS))
    (NEQ TEDPROC (THIS.PROCESS))
    (DEL.PROCESS TEDPROC)
    (TEDIT.DEACTIVATE.WINDOW TEDW)]

```

(TEDIT.MAPLINES

```

[LAMBDA (TEXTOBJ FN)
; Edited 29-May-91 18:19 by jds

(* Go thru the visible lines in a textobj and call a mapping fn on them)

(* FN has 2 args%: the LINEDESCRIPTOR, and a VISIBLEFLG to say if the line is visible on the screen.)

(bind (LINE _ (fetch (LINEDESCRIPTOR NEXTLINE) of (fetch (TEXTOBJ LINES) of TEXTOBJ)))
 (BOT _ (fetch (TEXTOBJ WBOTTOM) of TEXTOBJ))
 [TOP _ (fetch (REGION HEIGHT) of (DSPCLIPPINGREGION (\TEDIT.PRIMARYW TEXTOBJ) while LINE
do (COND
 ((EQ (APPLY* FN LINE (AND (ILESSP (fetch (LINEDESCRIPTOR YBOT) of LINE)
TOP)
(IGE (fetch (LINEDESCRIPTOR YBOT) of LINE)
BOT)))
' STOP)
(RETURN)))
 (SETQ LINE (fetch (LINEDESCRIPTOR NEXTLINE) of LINE)]

```

(TEDIT.MAPPIECES

```

[LAMBDA (TEXTOBJ FN FNARG)
; Edited 22-Apr-93 16:02 by jds

;; Go thru all the pieces in a document, applying a function to them serially
;; FN is a function of 3 args (PIECE CH#-of-1st-char-in-piece PIECE# in table FNARG)
(PROG ((PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
 (CH# 1)
 PCNODE PC)
 (SETQ PC (\GETBASEPTR (\FIRSTNODE PCTB)
 0))
 (OR (ATOM PC)
  (RETURN (for I from 1 while PC do [COND
 ((EQ (APPLY* FN CH# PC I FNARG)
' STOP)
 (RETURN (LIST CH# PC I)
 (add CH# (fetch (PIECE PLEN) of PC))
 (SETQ PC (fetch (PIECE NEXTPIECE) of PC)]

```

(TEDIT.MOVE

```

[LAMBDA (FROM TO)
; Edited 2-May-95 12:24 by sybalsky:mv:envos

;; Move the text described by the selection FROM to the place described by the selection TO
(SETQ TEDIT.MOVE.PENDING NIL)
; First, Turn off the global flag that got us here.
(COND
 ((NOT (AND FROM (fetch (SELECTION SET) of FROM)))
  ; There MUST be a source selected first.
  (TEDIT.PROMPTPRINT (fetch (SELECTION \TEXTOBJ) of TO)
   "Move source selection hasn't been set yet." T))

```

```

((ZEROP (fetch (SELECTION DCH) of FROM)) ; The source selection is empty. Just turn it off.
(\SHOWSEL FROM NIL NIL))
((fetch (TEXTOBJ TXTREADONLY) of (fetch (SELECTION \TEXTOBJ) of TO)) ; The target is read-only. Skip it..
(\SHOWSEL FROM NIL NIL))
(T (\SHOWSEL FROM NIL NIL) ; Before all else, make sure the copy source selection is turned
; off
(COND
  ((AND TO (fetch (SELECTION SET) of TO)) ; Can only do copy if there's a target selection
  (PROG ((TOOBJ (fetch (SELECTION \TEXTOBJ) of TO))
    (FROMOBJ (fetch (SELECTION \TEXTOBJ) of FROM))
    (TOLEN (fetch (TEXTOBJ TEXTLEN) of (fetch (SELECTION \TEXTOBJ) of TO)))
    (TOPCTB (fetch (TEXTOBJ PCTB) of (fetch (SELECTION \TEXTOBJ) of TO)))
    (FROMPCTB (fetch (TEXTOBJ PCTB) of (fetch (SELECTION \TEXTOBJ) of FROM)))
    (CROSSCOPY (NEQ (fetch (SELECTION \TEXTOBJ) of FROM)
      (fetch (SELECTION \TEXTOBJ) of TO)))
    LEN INSPC INSPC# PC NPC PCCH NPCCH OPLEN EVENT REPLACING INSERTCH# PCLST OBJ COPYFN
    UNDOCHAIN) ; Find the insertion point
    (SETQ PCLST (TEDIT.SELECTED.PIECES FROMOBJ FROM CROSSCOPY (FUNCTION \TEDIT.MOVE.PIECEMAPFN)
      FROMOBJ TOOBJ)) ; Grab the pieces that reflect the source selection
    (SETQ REPLACING (TEDIT.DO.BLUEPENDINGDELETE TO OBJ)) ; Do any blue-pending-delete
    (SETQ TOPCTB (fetch (TEXTOBJ PCTB) of TOOBJ)) ; Get the new PCTB and text length
    (SETQ EVENT (fetch (TEXTOBJ TXTHISTORY) of TOOBJ))
    (SETQ LEN (IDIFFERENCE (fetch (SELECTION CHLIM) of FROM)
      (fetch (SELECTION CH#) of FROM)))
    (\DELETECH (fetch (SELECTION CH#) of FROM)
      (fetch (SELECTION CHLIM) of FROM)
      (fetch (SELECTION DCH) of FROM)
      FROMOBJ) ; Now delete the text from its old place
    (\FIXDLINES (fetch (TEXTOBJ LINES) of FROMOBJ)
      FROM
      (fetch (SELECTION CH#) of FROM)
      (fetch (SELECTION CHLIM) of FROM)
      FROMOBJ)
    (SETQ TOLEN (fetch (TEXTOBJ TEXTLEN) of TOOBJ))
    (\SHOWSEL TO NIL NIL) ; NOW turn off the target selection.
    (replace (SELECTION SET) of FROM with NIL)
    [COND
      ((EQ (fetch (SELECTION POINT) of TO)
        'LEFT)
        (SETQ INSERTCH# (fetch (SELECTION CH#) of TO)))
      (T (SETQ INSERTCH# (fetch (SELECTION CHLIM) of TO)
        ; Figure out where to do the insertion.
        (\TEDIT.INSERT.PIECES TOOBJ INSERTCH# PCLST LEN NIL NIL CROSSCOPY)
        ; Get the pieces that actually got inserted, so we can UNDO the
        ; move
        ;; Keep the target from sharing a piece with type-in by accident:
        (replace (TEXTOBJ \INSERTPCVALID) of TOOBJ with NIL)
        ;; Keep \DELETECH from playing clever games with the piece if it's new type-in: Don't let it be reclaimed by the deletion:
        (replace (TEXTOBJ \INSERTPCVALID) of FROMOBJ with NIL)
        (replace (TEXTOBJ \INSERTPC) of FROMOBJ with NIL)
        (\TEDIT.HISTORYADD TOOBJ (create TEDIT.HISTORYEVENT
          THTEXTOBJ _ TOOBJ
          THACTION _ (COND
            (REPLACING 'ReplaceMove)
            (T 'Move))
          THLEN _ LEN
          THCH# _ INSERTCH#
          THFIRSTPIECE _ (LIST PCLST)
          THAUXINFO _ FROMOBJ
          THOLDINFO _ (fetch (SELECTION CH#) of FROM)))
          ; Make a history-list entry for the COPY.
          (replace (TEXTOBJ \DIRTY) of TOOBJ with T) ; Mark the document changed
          (replace (TEXTOBJ TEXTLEN) of TOOBJ with (IPLUS LEN TOLEN)) ; Set the new length
          (\FIXILINES TOOBJ TO INSERTCH# LEN TOLEN) ; Fix LINES and SELs
          (COND
            ((EQ (fetch (TEXTOBJ FORMATTEDP) of TOOBJ)
              (fetch (TEXTOBJ FORMATTEDP) of FROMOBJ))
              ; Either both of the files are formatted or neither is. This case is
              ; OK
            )
            ((fetch (TEXTOBJ FORMATTEDP) of TOOBJ) ; The source wasn't formatted, but the target is. Go convert the
              ; copied text.
              (\TEDIT.CONVERT.TO.FORMATTED TOOBJ INSERTCH# (IPLUS INSERTCH# LEN)))
            (T ; The source is formatted and the target isn't. Give the guy a
              ; choice.
              ; For now, convert the target file to formatted.
              (\TEDIT.CONVERT.TO.FORMATTED TOOBJ)))
            (TEDIT.UPDATE.SCREEN FROMOBJ)
            (TEDIT.UPDATE.SCREEN TOOBJ) ; Refresh the display
            (replace (SELECTION CH#) of TO with INSERTCH#) ; Correct the target selection
            (replace (SELECTION CHLIM) of TO with (IPLUS INSERTCH# LEN))

```



```

        (replace (SELECTION DCH) of TO with LEN)
        (replace (SELECTION DX) of TO with 0)
        (replace (SELECTION POINT) of TO with 'RIGHT)
        (COND
          ((NEQ TO FROM)
            (\FIXSEL FROM FROMOBJ)
            (\FIXSEL (fetch (TEXTOBJ SEL) of FROMOBJ)
              FROMOBJ))
          (\FIXSEL TO TOOBJ)
          (\SHOWSEL TO NIL T)))
      (T
        (TEDIT.PROMPTPRINT (fetch (SELECTION \TEXTOBJ) of FROM)
          "Please select a destination for the MOVE first." T])
        ; There is no target selection -- complain

```

(TEDIT.QUIT

```

[LAMBDA (STREAM VALUE)
  ; Edited 12-Jun-90 17:49 by mitani
  ;; Force the edit session supported by STREAM to terminate, and to return VALUE
  (COND
    ((type? STREAM STREAM)
      (SETQ STREAM (fetch (TEXTSTREAM TEXTOBJ) of STREAM)))
    ((type? TEXTOBJ STREAM)
      ; If he gave us a textofd, get the textobj
      ; A Textobj is just fine
    )
    (T
      (\ILLEGAL.ARG STREAM)))
  (replace (TEXTOBJ EDITFINISHEDFLG) of STREAM with (OR VALUE T))
  ; tell the command loop to stop next time through
  (PROG (MAINW)
    (COND
      ([AND (fetch (TEXTOBJ \WINDOW) of STREAM)
        (NEQ (SETQ MAINW (\TEDIT.PRIMARYW STREAM))
          (PROCESSPROP (TTY.PROCESS)
            'WINDOW])
        ; there is a main window of the stream, and it is not the window of the tty process, so give it the tty
        (TTY.PROCESS (WINDOWPROP MAINW 'PROCESS))
        (AND (NEQ (TTY.PROCESS)
          (THIS.PROCESS))
          (until [OR (NOT (WINDOWPROP MAINW 'PROCESS))
            (PROCESS.FINISHEDP (WINDOWPROP MAINW 'PROCESS))
            ; Wait until the Edit process has had a chance to go away before
            ; continuing here.
          ]
            do
              (DISMISS]))
    )

```

(TEDIT.STRINGWIDTH

```

[LAMBDA (STR FONT TERMSA)
  (COND
    (TERMSA
      (* jds "19-AUG-83 14:40")
      (* We have a terminal table to take account of.
        Do so.)
      (for CH instring STR sum (TEDIT.CHARWIDTH CH FONT TERMSA)))
    (T
      (* Just use the native character widths)
      (for CH instring STR sum (SELCHARQ CH
        (TAB 36)
        (CHARWIDTH CH FONT)))
    )

```

(TEDIT.INSERT

```

[LAMBDA (CH SEL STREAM)
  ; Edited 29-May-91 18:22 by jds
  (* Insert the character CH at the appropriate spot in the text.)
  (DECLARE (LOCALVARS . T))
  (PROG [(TEXTOBJ (COND
    ((type? STREAM STREAM)
      (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
    (T STREAM]
    (* If we got a STREAM, change it into a textobj))
    (COND
      ((NOT (AND SEL (fetch (SELECTION SET) of SEL)))
        (TEDIT.PROMPTPRINT TEXTOBJ "Please select a place for the insertion." T)
        (RETURN)))
      (COND
        ((fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
          (* There is a window; make sure the insert point is on-screen, and turn off any highlighted selection)
          (TEDIT.NORMALIZECARET TEXTOBJ SEL)
          (\SHOWSEL SEL NIL NIL)))
        (PROG ((CH# (TEDIT.GETPOINT STREAM SEL))
          (XPOINT (SELECTQ (fetch (SELECTION POINT) of SEL)
            (LEFT (fetch (SELECTION X0) of SEL))
            (RIGHT (fetch (SELECTION XLIM) of SEL))
            NIL))
          (OTEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)))
          (SELCHARQ CH
            ((CR %#^M 1,CR)
              (\INSERTCR CH CH# TEXTOBJ))
            (* This was a CR. Go do the para breaking as needed)

```

```
(\INSERTCH CH CH# TEXTOBJ))
(\FIXILINES TEXTOBJ SEL CH# 1 OTEXTLEN)
(TEDIT.INSERT.UPDATESCREEN CH CH# 1 XPOINT TEXTOBJ SEL OTEXTLEN NIL NIL NIL T))
```

(TEXTOBJ

```
[LAMBDA (STREAM)
  (COND
    ((type? TEXTOBJ STREAM)
     STREAM)
    ((AND (type? STREAM STREAM)
      (fetch (TEXTSTREAM TEXTOBJ) of STREAM)
      (type? TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))))
     (* It's a TEXTSTREAM)
    (fetch (TEXTSTREAM TEXTOBJ) of STREAM)
    ((AND (PROCESSP STREAM)
      (PROCESS.WINDOW STREAM))
     (* It's an edit PROCESS)
    (WINDOWPROP (PROCESS.WINDOW STREAM)
      'TEXTOBJ))
    [(AND (WINDOWP STREAM)
      (WINDOWPROP STREAM 'TEXTOBJ)]
    [(AND (DISPLAYSTREAMP STREAM)
      (WINDOWPROP STREAM 'TEXTOBJ)]
    ((\ILLEGAL.ARG STREAM))
```

(TEXTSTREAM

```
[LAMBDA (STREAM)
  (COND
    ((AND (type? STREAM STREAM)
      (fetch (TEXTSTREAM TEXTOBJ) of STREAM)
      (type? TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))))
     (* It's a stream, and is really a TEXT stream.
      Just return it.)
    STREAM)
    ((type? TEXTOBJ STREAM)
     (fetch (TEXTOBJ STREAMHINT) of STREAM))
    ((AND (PROCESSP STREAM)
      (PROCESS.WINDOW STREAM))
     (* It's an edit process, so grab the text stream from the edit
      window.)
    (WINDOWPROP (PROCESS.WINDOW STREAM)
      'TEXTSTREAM))
    [(AND (WINDOWP STREAM)
      (WINDOWPROP STREAM 'TEXTSTREAM)]
    [(AND (DISPLAYSTREAMP STREAM)
      (WINDOWPROP STREAM 'TEXTSTREAM)]
    ((\ILLEGAL.ARG STREAM)
    ])
  ])
```

(TEDIT.INCLUDE

```
[LAMBDA (TEXTOBJ FILE START END)
  (* A NATIVE text includer%: Includes part of a file, without checking to see if it's a bravo file, a TEdit file or whatever.)
  (* (PROG ((LEN (IDIFFERENCE
    (OR END (GETEOFPTR FILE)) (OR START 0)))
    (SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)) NPC)
    (SETQ NPC (create PIECE PFILE
    (GETOFD FILE (QUOTE INPUT)) PFPOS _
    (OR START 0) PLEN LEN PLOOKS
    (TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL)
    PPARALOOKS NIL)) (* Create a PIECE to describe the text)
    (TEDIT.INSERT.PIECES TEXTOBJ
    (fetch (SELECTION CH#) of SEL) NPC LEN)
    (* Insert it in the document) (add (fetch
    (TEXTOBJ TEXTLEN) of TEXTOBJ) LEN)
    (* And update the document's length)
    (TEDIT.MARK.LINES.DIRTY TEXTOBJ
    (fetch (SELECTION CH#) of SEL)
    (IPLUS (fetch (SELECTION CH#) of SEL) LEN))
    (* Mark the screen dirty, so updating it will find something to do)
    (replace (SELECTION CHLIM) of SEL with
    (IPLUS (fetch (SELECTION CH#) of SEL) LEN))
    (replace (SELECTION DCH) of SEL with LEN)
    (replace (SELECTION DX) of SEL with 0)
    (replace (SELECTION POINT) of SEL with
    (QUOTE RIGHT)) (replace (SELECTION SELKIND) of SEL with
    (QUOTE CHAR)) (replace (SELECTION SELOBJ) of SEL with
    NIL) (COND ((fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
    (\SHOWSEL SEL NIL NIL) (TEDIT.UPDATE.SCREEN
    TEXTOBJ) (\FIXSEL SEL TEXTOBJ)
    (\SHOWSEL SEL NIL T))) (* Update the screen)
    (replace (TEXTOBJ \DIRTY) of TEXTOBJ with T)
    (SETUPGETCH (fetch (SELECTION CH#) of SEL) TEXTOBJ)))
```

(HELP)]

(\TEDIT.INSERT.PIECES

[LAMBDA (TEXTOBJ CH# FIRSTPIECE %#CHARS INSPC INSPC# CROSSCOPY DONTDIRTY COPYING)

; Edited 4-May-95 08:16 by sybalsky:mv:envos

;; Inserts a series of pieces into TEXTOBJ in front of character CH#.

;; If FIRSTPIECE is a PIECE, this will follow the next-piece pointer chain; if FIRSTPIECE is a list, it is a list of pieces to insert.

;; If CROSSCOPY is non-NIL, the pieces' contents will be copied, to preserve text in case the original is deleted.

;; INSPC and INSPC# are accelerators for where in the PCTB the new pieces should go.

;; DONTDIRTY is T if this is a change not visible to the user--one that shouldn't "dirty" the document. This is used for NS-character encoding recognition during line formatting.

;; COPYING is T if these pieces are being inserted by a COPY operation. This lets us call the AFTERCOPYFN on image objects.

;; It is the CALLER'S RESPONSIBILITY to make sure the pieces to be inserted are 'safe' --that they are, if necessary, copies of the originals, and can safely be modified. ; NB THAT THIS DOES NOT UPDATE TEXTLEN

(COND

((OR DONTDIRTY (NOT (fetch (TEXTOBJ TXTREADONLY) of TEXTOBJ)))

;; Only do this if you're allowed to change the document, or it's a TEdit-internal fixup change, as for NS char recognition.

(LET ((TOLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))

(TOPCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))

(CURCH# CH#)

LEN PC PREVPC NPC UNDOCHAIN PSTR SRCPPFILE START-OF-PIECE)

(DECLARE (SPECVARS START-OF-PIECE))

; Get a handle on the piece we're to insert within or in front of
(* COND ((ZEROP (fetch (BTREENODE TOTLEN) of TOPCTB))
(* ; "PCTB is empty." (\INSERT.FIRST.PIECE TEXTOBJ)))

(SETQ INSPC (\CHTOPC CH# TOPCTB T))

; And the piece, itself. (Used to be (OR INSPC (\CH...)), but we
; MUST set START-OF-PIECE, so must make the call to
; \CHTOPC.

(replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL)

; Force later insertions to make new pieces.

[COND

((IGREATERP CH# TOLEN)

; We're inserting at end of file; leave the piece to insert before as
; LASTPIECE

) ((IEQP CH# START-OF-PIECE)

; The insertion is IN FRONT of this piece; just continue on

) (T (SETQ INSPC (\SPLITPIECE INSPC (- CH# START-OF-PIECE)
TEXTOBJ])

; Nope, we're inserting INSIDE this piece. Split it in two.

(COND

((NEQ INSPC 'LASTPIECE)

; Not the last piece, so back up using the pointer.

(SETQ PREVPC (fetch (PIECE PREVPiece) of INSPC)))

(NOT (ZEROP (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)))

; If we are at the end, AND there is text before us, find it thru the
; pctx.(SETQ PREVPC (\CHTOPC (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
TOPCTB)))(T (SETQ PREVPC NIL))) ; Otherwise, there is no piece before where we're inserting.
; For pushing para looks in

(bind [PC _ (create PIECE using (COND

((LISTP FIRSTPIECE)

(pop FIRSTPIECE))

(T FIRSTPIECE]

(LEN _ 0)

(PCCOUNT _ 0) first (SETQ UNDOCHAIN PC) while (AND PC (OR (NOT %#CHARS)

(ILESSP LEN %#CHARS)))

do ; Now insert the copied pieces into the new place

(COND

((AND CROSSCOPY (SETQ SRCPPFILE (fetch (PIECE PFILE) of PC)))

;; If this is a cross-document copy, and the text comes from a file, we must REALLY make a copy of the text, lest the
;; source file be deleted.

;; (replace PSTR of PC with (SETQ PSTR (ALLOCSTRING (fetch PLEN of PC) NIL NIL (fetch PFATP of PC))))

(replace (PIECE PFILE) of PC with (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW))

; Create the holding file

[COND

((NOT (OPENP SRCPPFILE))

; The source file was CLOSED -- reopen it, for our us

(replace (PIECE PFILE) of PC with (SETQ SRCPPFILE (OPENSTREAM SRCPPFILE 'INPUT
'OLD]

(SETFILEPTR SRCPPFILE (fetch (PIECE PFPOS) of PC))

[COPYCHARS SRCPPFILE (fetch (PIECE PFILE) of PC)

(fetch (PIECE PFPOS) of PC)

(IPLUS (fetch (PIECE PFPOS) of PC)

(COND

((fetch (PIECE PFATP) of PC)

(LLSH (fetch (PIECE PLEN) of PC)

1))

(T (fetch (PIECE PLEN) of PC]

(replace (PIECE PFPOS) of PC with 0)))

(replace (PIECE PLOOKS) of PC with (\TEDIT.UNIQUIFY.CHARLOOKS (fetch (PIECE PLOOKS) of PC)
TEXTOBJ))

```

(replace (PIECE PPARALOOKS) of PC with (\TEDIT.UNIQIFY.PARALOOKS (fetch (PIECE PPARALOOKS)
                                                                    of PC)
TEXTOBJ))
; Assume that the new document knows about this piece's looks

[COND
  ((NULL FIRSTPIECE)
   (SETQ NPC NIL))
  [(LISTP FIRSTPIECE)
   (SETQ NPC (create PIECE using (pop FIRSTPIECE)
                                ; If the piece list really IS a list, grab the next piece from the front
                                ; Otherwise, follow the NEXTPIECE chain among pieces
                                (SETQ NPC (create PIECE using (fetch (PIECE NEXTPIECE) of PC]
                                ; Insert the piece into the new document
                                (\INSERTPIECE PC INSPC TEXTOBJ NIL)
                                [COND
                                  (COPYING
                                   ;; For objects, call the optional AFTERCOPYFN.
                                   (LET (OBJ AFTERN)
                                    (AND (SETQ OBJ (ffetch (PIECE POBJ) of PC))
                                     (SETQ AFTERN (IMAGEOBJPROP OBJ 'AFTERCOPYFN))
                                     (APPLY* AFTERN OBJ PC CURCH#]
                                   (add CURCH# (fetch (PIECE PLEN) of PC))
                                   (add LEN (fetch (PIECE PLEN) of PC))
                                   (SETQ PC NPC))
                                   (\TEDIT.DIFFUSE.PARALOOKS PREVPC INSPC)
                                   UNDOCHAIN])

```

(\TEDIT.MOVE.PIECEMAPFN

[LAMBDA (PC TEXTOBJ FROMOBJ TOOBJ)

; Edited 12-Jun-90 17:50 by mitani
(* Called by TEDIT.MOVE via TEDIT.SELECTED.PIECES, to
do the move-operation processing on the candidate pieces.)

```

(PROG (OBJ MOVEFN)
  (SETQ PC (create PIECE using PC PNEW _ T))
  [COND
    [(fetch (PIECE POBJ) of PC)
     (* No matter what, we need a fresh copy.)
     (* This piece describes an object
      * Call its WHENMOVEDFN.)
     (SETQ OBJ (fetch (PIECE POBJ) of PC))
     (COND
      ((SETQ MOVEFN (IMAGEOBJPROP OBJ 'WHENMOVEDFN)) (* If there's an eventfn for moving, use it.)
       (APPLY* MOVEFN OBJ (CAR (fetch (TEXTOBJ \WINDOW) of FROMOBJ)
                             (fetch (TEXTOBJ STREAMHINT) of FROMOBJ)
                             (fetch (TEXTOBJ STREAMHINT) of TOOBJ)
                             (fetch (PIECE PSTR) of PC)
                             (* If the piece is a string, make our own copy of the string header, even tho we share characters.)
                             (replace (PIECE PSTR) of PC with (SUBSTRING (fetch (PIECE PSTR) of PC)
                                                                    1
                                                                    (fetch (PIECE PLEN) of PC)
                                                                    (RETURN PC]))

```

(\TEDIT.OBJECT.SHOWSEL

[LAMBDA (TEXTOBJ SEL ON SELWINDOW)

; Edited 12-Jun-90 17:50 by mitani

;; We are highlighting (or dehighlighting) a selected object. Let it know.

```

(LET ((X (fetch (SELECTION X0) of SEL))
      (Y (fetch (SELECTION Y0) of SEL))
      (FIRSTLINE (CAR (fetch (SELECTION L1) of SEL)))
      (OBJ (fetch (SELECTION SELOBJ) of SEL))
      (WIDTH (fetch (SELECTION DX) of SEL))
      (XOFFSET (DSPXOFFSET NIL SELWINDOW))
      (YOFFSET (DSPYOFFSET NIL SELWINDOW))
      (IMAGEFN (IMAGEOBJPROP (fetch (SELECTION SELOBJ) of SEL)
                                'WHENOPERATEDONFN))
      (WWIDTH (WINDOWPROP SELWINDOW 'WIDTH))
      (WHEIGHT (WINDOWPROP SELWINDOW 'HEIGHT))
      IMAGEBOX)
  (COND
    ((INSIDE? (CREATEREGION 0 0 WWIDTH WHEIGHT)
              X Y)
     (SETQ IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
                       (APPLY* (IMAGEOBJPROP OBJ 'IMAGEBOXFN)
                                OBJ SELWINDOW)))
     [COND
       (FIRSTLINE
        ;; There's really a line this selection is being displayed on, so we need to use the YBASE of the line- the object's descent,
        ;; rather than the YBOT, which is what Y0 is.
        (SETQ Y (- (fetch (LINEDESCRIPTOR YBASE) of FIRSTLINE)
                   (fetch (IMAGEBOX YDESC) of IMAGEBOX]
        (RESETLST
         [RESETSAVE (DSPXOFFSET (IDIFFERENCE (IPLUS X XOFFSET)
                                              (fetch XKERN of IMAGEBOX))
                     SELWINDOW)
         (LIST (FUNCTION DSPXOFFSET)
               XOFFSET

```

```

(WINDOWPROP SELWINDOW 'DSP]
(RESETSAVE (DSPYOFFSET (IPLUS Y YOFFSET)
              SELWINDOW)
  (LIST (FUNCTION DSPYOFFSET)
        YOFFSET SELWINDOW))
(RESETSAVE (DSPCLIPPINGREGION (create REGION
                                LEFT _ 0
                                BOTTOM _ 0
                                WIDTH _ (IMIN WIDTH (IDIFFERENCE (fetch (TEXTOBJ WRIGHT)
                                                                    of TEXTOBJ)
                                                                    X))
                                HEIGHT _ (fetch YSIZE of IMAGEBOX))
              SELWINDOW)
  (LIST (FUNCTION DSPCLIPPINGREGION)
        (DSPCLIPPINGREGION NIL SELWINDOW)
        SELWINDOW))
[AND IMAGEFN (ERSETQ (APPLY* IMAGEFN OBJ SELWINDOW (COND
                                                        (ON 'HIGHLIGHTED)
                                                        (T 'UNHIGHLIGHTED))
  SEL
  (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ))]]

```

(\TEDIT.RESTARTFN

```

[LAMBDA (TEXT WINDOW PROPS)
  (replace (TEXTOBJ \WINDOW) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT) with NIL)
  (PROG [(ODIRTY (fetch (TEXTOBJ \DIRTY) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
                (SETQ TEXT (OPENTEXTSTREAM TEXT WINDOW NIL NIL PROPS))
                (replace (TEXTOBJ \DIRTY) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT) with ODIRTY))
    (\TEDIT.COMMAND.LOOP (fetch (TEXTSTREAM TEXTOBJ) of TEXT))
    (CLOSEW WINDOW)
    (\TEXTCLOSEF TEXT)
    (replace (STREAM ACCESSBITS) of TEXT with BothBits)
    (AND (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
                  'AFTERQUITFN)
      (APPLY* (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
                        'AFTERQUITFN)
        WINDOW TEXT))
  ])

```

; Edited 12-Jun-90 17:51 by mitani
 (* Restarts a TEdit session.)
 (* Unattach the window, so we do a redisplay.)
 (* Now reconnect the world together again)
 (* Run the editing engine)
 (* Close the edit window)
 (* Close the underlying files)
 (* But leave the stream itself accessible)
 (* Apply any post-window-close (and post-QUIT) function)

(\TEDIT.CHARDELETE

```

[LAMBDA (TEXTOBJ SCRATCHSTRING SEL)
  ;; Do character-backspace deletion for TEDIT
  (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
  (PROG ((PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
    TLEN INSCH# INSPC INSPC# TLOOKS START-OF-PIECE)
    (COND
      [NIL [NOT (ZEROP (SETQ TLEN (fetch (STRINGP OFFST) of SCRATCHSTRING)
        (replace (STRINGP OFFST) of SCRATCHSTRING with (SUB1 TLEN))
        (replace (STRINGP LENGTH) of SCRATCHSTRING with (ADD1 (fetch (STRINGP LENGTH) of SCRATCHSTRING)
          (T (SELECTQ (fetch (SELECTION POINT) of SEL)
            (LEFT (SETQ INSCH# (SUB1 (fetch (SELECTION CH#) of SEL))))
            (RIGHT (SETQ INSCH# (SUB1 (fetch (SELECTION CHLIM) of SEL))))
            NIL)
          (COND
            ((ILEQ INSCH# 0)
              (RETURN))
            ;; (SETQ INSPC (\EDITELT PCTB (ADD1 (SETQ INSPC# (\CHTOPCNO INSCH# PCTB))))
            (SETQ INSPC (\CHTOPC INSCH# PCTB T))
            (SETQ TLOOKS (\TEDIT.APPLY.STYLES (fetch (PIECE PLOOKS) of INSPC)
              INSPC TEXTOBJ))
            [while (AND INSPC (fetch CLINVISIBLE of TLOOKS))
              do
                (SETQ INSPC (fetch (PIECE PREVPIECE) of INSPC))
                (SETQ INSCH# (SUB1 START-OF-PIECE))
                (add START-OF-PIECE (IMINUS (fetch (PIECE PLEN) of INSPC)))
                (COND
                  (INSPC (SETQ TLOOKS (\TEDIT.APPLY.STYLES (fetch (PIECE PLOOKS) of INSPC)
                    INSPC TEXTOBJ))
                (COND
                  ((ILEQ INSCH# 0)
                    (RETURN))
                  (NOT (fetch CLPROTECTED of TLOOKS))
                  (replace (SELECTION CHLIM) of SEL with (ADD1 (replace (SELECTION CH#) of SEL with INSCH#)))
                  (replace (SELECTION DCH) of SEL with 1)
                  (\SHOWSEL SEL NIL NIL)
                ))
                ; Can't backspace past start of document
                ; If we didn't really insert the text yet, just remove from the text to
                ; be inserted
                ; Delete the character just before the current insertpoint.
                ; Back over any invisible text, which we're no allowed to delete.
                ; We backed up to the start of the document. Can't go no further.
                ; Can only backspace if the char to go isn't protected.
                ; Set up the selection to point to the character which is to be
                ; deleted.
                ; Turn off the underlining, if any, so there's no garbage.

```

```

(\FIXSEL SEL TEXTOBJ) ; Fix the selection up so it points to the right line and all
(\TEDIT.DELETE SEL TEXTOBJ T) ; And delete it.

])

```

(\TEDIT.COPY.PIECEMAPFN

[LAMBDA (PC TEXTOBJ FROMOBJ TOOBJ)

; Edited 12-Jun-90 17:50 by mitani

(* Called by TEDIT.COPY via TEDIT.SELECTED.PIECES, to do

the copy-operation processing on the candidate pieces.)

(PROG (OBJ NEWOBJ COPYFN)

(SETQ PC (create PIECE using PC PNEW _ T))

(* No matter what, we need a fresh copy.)

[COND

((fetch (PIECE POBJ) of PC)

(* This piece describes an object)

(SETQ OBJ (fetch (PIECE POBJ) of PC))

[COND

[(SETQ COPYFN (IMAGEOBJPROP OBJ 'COPYFN))

(SETQ NEWOBJ (APPLY* COPYFN OBJ (fetch (TEXTOBJ STREAMHINT) of FROMOBJ)

(fetch (TEXTOBJ STREAMHINT) of TOOBJ)))

(COND

((EQ NEWOBJ 'DON'T)

(* He said not to copy this piece --
abort the whole copy.)

(TEDIT.PROMPTPRINT TEXTOBJ "COPY of this object not allowed." T)

(RETFROM 'TEDIT.COPY))

(NEWOBJ (replace (PIECE POBJ) of PC with NEWOBJ))

(T (replace (PIECE POBJ) of PC with (COPYALL OBJ)

(OBJ

(* No copy fn; just strike off a copy of our own)

(replace (PIECE POBJ) of PC with (COPYALL OBJ)

(COND

((SETQ COPYFN (IMAGEOBJPROP OBJ 'WHENCOPIEDFN)) (* If there's an eventfn for copying, use it.)

(APPLY* COPYFN OBJ (WINDOWPROP (CAR (fetch (TEXTOBJ \WINDOW) of TOOBJ))

'DSP)

(fetch (TEXTOBJ STREAMHINT) of FROMOBJ)

(fetch (TEXTOBJ STREAMHINT) of TOOBJ]

[COND

((fetch CLPROTECTED of (fetch (PIECE PLOOKS) of PC)) (* The source text was protected;
unprotect the copy.)(replace (PIECE PLOOKS) of PC with (\TEDIT.UNIQUIFY.CHARLOOKS (create CHARLOOKS
using (fetch (PIECE PLOOKS)
of PC)CLPROTECTED _ NIL CLSELHERE _
NIL)

TOOBJ]

(RETURN PC])

(\TEDIT.DELETE

[LAMBDA (SEL STREAM SELOFF)

; Edited 29-May-91 18:22 by jds

; DELETE THE CHARACTERS SPECIFIED FROM THE MAIN

; TEXT.

; SELOFF => The selection is already turned off.

(LET* ((TEXTOBJ (TEXTOBJ STREAM))

(CH# (fetch (SELECTION CH#) of SEL))

(CHLIM (fetch (SELECTION CHLIM) of SEL))

(LINES (fetch (TEXTOBJ LINES) of TEXTOBJ))

(WINDOW (fetch (TEXTOBJ \WINDOW) of TEXTOBJ))

(HEIGHTCHANGED NIL)

(NLINE1 NIL)

(CRFLAG NIL)

(LINES\DELETED NIL)

OLINE1 OLINEN LEN NEXTLINE NL OLINE DX OCHLIM OXLIM OLHEIGHT OLASCENT OLDESCENT DY PREVLIN TEXTLEN

OCR\END SAVEDWIDTH IMAGECACHE)

[SETQ LEN (COND

((IGREATERP CH# (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))

; Past end of text, so don't delete any

0)

((IGEQL CH# CHLIM)

; Start is past end, so don't delete any.

0)

((ZEROP (fetch (SELECTION DCH) of SEL))

; Just a caret--no text really selected--so don't delete any

0)

((ZEROP CHLIM)

; CHLIM is before start of text, so don't delete any

0)

(T

; The normal case.

(IDIFFERENCE CHLIM CH#)

; # of characters to be deleted

(COND

((OR (fetch (TEXTOBJ TXTREADONLY) of TEXTOBJ)

(NOT (fetch (SELECTION SET) of SEL))

(ZEROP LEN))

; If the selection isn't set, OR the document is read-only, OR the
selection contains no characters, don't do anything.

)

(T (AND WINDOW (TEDIT.NORMALIZECARET TEXTOBJ SEL)) ; If the text appears in a window, move the deletion point
; on-screen

(SETQ OLINE1 (fetch (SELECTION L1) of SEL))

(SETQ OLINEN (fetch (SELECTION LN) of SEL))

(\TEDIT.SHOWSELS TEXTOBJ NIL NIL)

; Turn off the selection's highlighting

(AND LINES (\FIXDLINES LINES SEL CH# CHLIM TEXTOBJ))

```

; Update the line descriptors to account for the deletion
; Do the actual deletion of characters
(replace THPOINT of (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ with (fetch (SELECTION POINT)
of SEL))
; Remember which side of the selection we were on, in case it
; gets undone.
(replace (SELECTION CH#) of SEL with (IMAX 1 CH#))
(replace (SELECTION CHLIM) of SEL with (fetch (SELECTION CH#) of SEL))
(replace (SELECTION POINT) of SEL with 'LEFT)
(replace (SELECTION DCH) of SEL with 0)
(COND
  (WINDOW
    (SETQ TEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))
    ; If there's no window to update, don't bother
    ; The new text length
    (for OLINE1 inside (fetch (SELECTION L1) of SEL) as OLINE1
      inside (fetch (SELECTION LN) of SEL) as TOPLINE inside (fetch (TEXTOBJ LINES)
of TEXTOBJ)
      as THISW inside WINDOW
      do (SETQ LINES\DELETED
        (\TEDIT.CLOSEUPLINES TEXTOBJ
          (OR (AND OLINE1 (COND
            ((fetch (LINEDESCRIPTOR DELETED) of OLINE1)
              (fetch (LINEDESCRIPTOR PREVLIN) of OLINE1))
            (T OLINE1)))
          (COND
            ([AND (fetch (LINEDESCRIPTOR NEXTLINE) of TOPLINE)
              (OR (IGEQ (fetch (LINEDESCRIPTOR CHAR1)
of (fetch (LINEDESCRIPTOR NEXTLINE)
of TOPLINE))
              (fetch (SELECTION CHLIM) of SEL))
              (ILESSP (fetch (LINEDESCRIPTOR CHARLIM)
of (fetch (LINEDESCRIPTOR NEXTLINE)
of TOPLINE))
              (fetch (SELECTION CH#) of SEL])
              ; The first line on the screen is already past where we're to
              ; delete. DON'T delete any lines
              NIL)
            (T TOPLINE)))
          (AND OLINE1 (COND
            ((fetch (LINEDESCRIPTOR DELETED) of OLINE1)
              (fetch (LINEDESCRIPTOR NEXTLINE) of OLINE1))
            (T OLINE1)))
          NIL THISW)))
      ; Remove any lines which were completely deleted.
      ;; This line must needs be reformatted the hard way--it isn't a left ragged line or one of the lines is off-screen.
      (replace (SELECTION DX) of SEL with 0)
      (TEDIT.UPDATE.SCREEN TEXTOBJ NIL T) ; Correct the text that's displayed already
      (\FIXSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
        TEXTOBJ) ; Then fix up the selection as needed.
      (\TEDIT.SHOWSELS TEXTOBJ NIL T))

```

(\TEDIT.DIFFUSE.PARALOOKS

[LAMBDA (PRIORPC SUCCEEDINGPC)

; Edited 12-Jun-90 17:48 by mitani

(* Given a discontinuity in paragraph looks, caused by an insertion or by a deletion%: Diffuse the existing paragraph looks across the discontinuity, so that all the pieces in a single paragraph have consistent looks. Give preference to diffusion toward the END of the document. This means that if you delete a CR between paragraphs, the second para is absorbed into the first.)

(* PRIORPC and SUCCEEDINGPC are the PIECES that bound the area of potential discontinuity%: the change will occur at one boundary or the other....)

```

(COND
  ((AND PRIORPC (NOT (fetch (PIECE PPARALAST) of PRIORPC))) (* The discontinuity is inside a paragraph.
Must copy para looks forward into the text.)
    (bind (PPLOOKS _ (fetch (PIECE PPARALOOKS) of PRIORPC))
      (PC _ (fetch (PIECE NEXTPIECE) of PRIORPC)) while PC do
        (* Copy para looks info in from the left, up the the first para
        break.)
        (replace (PIECE PPARALOOKS) of PC with PPLOOKS)
        (COND
          ((fetch (PIECE PPARALAST) of PC)
            (* If this piece ends a paragraph, we're done.)
            (RETURN)))
          (SETQ PC (fetch (PIECE NEXTPIECE) of PC))
        )
      )
    (COND
      ((AND SUCCEEDINGPC (NEQ SUCCEEDINGPC 'LASTPIECE))
        (* Only copy para looks in from the right if there is text to the right.)
        (bind (PPLOOKS _ (fetch (PIECE PPARALOOKS) of SUCCEEDINGPC))
          (PC _ (fetch (PIECE PREVPIECE) of SUCCEEDINGPC)) while (NEQ PC PRIORPC)
          do
            (* Copy para looks in from the right, up to the first para break)
            (COND
              ((fetch (PIECE PPARALAST) of PC)
                (* If this piece ends a paragraph, we're done.)
                (RETURN)))
            )
          )

```

```
(replace (PIECE PPARALOOKS) of PC with PPLOOKS)
(SETQ PC (fetch (PIECE PREVPiece) of PC))
```

(\TEDIT.FOREIGN.COPY?

[LAMBDA (SEL)

; Edited 21-Jan-93 11:46 by jds

;; IF the current process's window isn't a TEdit window, do a 'Copy' by BKSYSBUFing the selected text. Then turn off all the various indicators.

```
(PROG (PROCW (SOURCE.TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
      CH STREAM DEST.TEDIT? DEST.TEXTOBJ)
  [SETQ DEST.TEDIT? (AND (SETQ PROCW (PROCESSPROP (TTY.PROCESS)
                                                    'WINDOW))
                        (SETQ DEST.TEXTOBJ (WINDOWPROP PROCW 'TEXTOBJ))
                        (NOT (TEXTPROP DEST.TEXTOBJ 'COPYBYBKSYSBUF))
```

```
; Treat the destination specially if (1) the recipient process has a
; window, and (2) it's a TEdit window, and (3) the TEdit isn't
; declining special treatment by having COPYBYBKSYSBUF set
; in its props.
```

(COND

```
((ZEROP (fetch (SELECTION DCH) of SEL))
 (SETQ TEDIT.COPY.PENDING NIL))
```

; Nothing to copy (0 characters in selection); don't bother.

```
((IGREATERP (fetch (SELECTION CH#) of SEL)
 (FETCH (TEXTOBJ TEXTLEN) OF SOURCE.TEXTOBJ))
```

```
; Trying to copy from beyond the end of the document; don't
; bother
```

```
(SETQ TEDIT.COPY.PENDING NIL))
((OR (NOT DEST.TEDIT?)
```

```
(AND PROCW DEST.TEXTOBJ (NEQ SOURCE.TEXTOBJ DEST.TEXTOBJ)
 (fetch (TEXTOBJ EDITOPACTIVE) of DEST.TEXTOBJ)))
```

; OK -- receiver isn't TEdit. Do it the hard way.

[COND

```
[(AND (WINDOWPROP [OR PROCW (WFROMDS (PROCESS.TTY (TTY.PROCESS)
                                         'COPYINSERTFN))
```

(PROGN

; This is the exit for looked-string objects

(OBJECTOUTOFTEDIT SOURCE.TEXTOBJ SEL]

(T

```
; Old tedit method, run if OBJECTOUTOFTEDIT is NIL (ie., not
; installed yet)
```

;; Still used because COPYINSERT does (PRIN2 BKSYSBUF) if there's no insertfn, which creates undesired string quotes.

```
(\SETUPGETCH (fetch (SELECTION CH#) of SEL)
 SOURCE.TEXTOBJ)
```

; Go to the first character to be copied

```
(SETQ STREAM (fetch (TEXTOBJ STREAMHINT) of SOURCE.TEXTOBJ))
(for i from 1 to (fetch (SELECTION DCH) of SEL) do
```

```
;; Run thru the selected text, copying only those items that really ARE
;; characters--IMAGEOBJS don't get copied by this route.
```

(COND

```
((FIXP (SETQ CH (\BIN STREAM)))
 (BKSYSBUF (CHARACTER CH)))
```

(T (COPYINSERT CH]

; Then reset the copy-pending flags.

(\SHOWSEL SEL NIL NIL)

(SETQ TEDIT.COPY.PENDING NIL])

(\TEDIT.QUIT

[LAMBDA (W NOFORCE)

; Edited 12-Jun-90 17:50 by mitani

(* Called by the default TEDIT.DEFAULT.MENUFN to perform the QUIT command.)

```
(PROG* ((TEXTOBJ (WINDOWPROP W 'TEXTOBJ))
        (QUITFNS (TEXTPROP TEXTOBJ 'QUITFN))
        (QUITFLG RESP)
```

```
[for QUITFN inside QUITFNS while (AND (NEQ QUITFLG 'DON'T)
                                         (NEQ QUITFLG T))
```

do (COND

((EQ QUITFN T)

(SETQ QUITFLG T))

(T (AND QUITFN (NEQ QUITFN T)

(SETQ QUITFLG (APPLY* QUITFN W (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)

TEXTOBJ

(fetch (TEXTOBJ EDITPROPS) of TEXTOBJ]

(COND

((EQ QUITFLG 'DON'T)

(* The user supplied a QUITFN, and it returned "DON'T" %, so just ignore all this Fooferaw and keep editing.)

(RETURN))

[(AND (fetch (TEXTOBJ \DIRTY) of TEXTOBJ)

(NOT (fetch (TEXTOBJ MENUFLG) of TEXTOBJ))

(NEQ QUITFNS T)

(NEQ QUITFLG T))

(* If this document has changed, check with the user to make sure he really wants to do it.)

```
(replace (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ with (MOUSECONFIRM "Not saved yet; LEFT to Quit
anyway." T (fetch (TEXTOBJ PROMPTWINDOW)
of TEXTOBJ]
```

(T

```
(* Go ahead and quit the next time we see the main command
loop.)
```



```

      (replace (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ with T)))
[AND (fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
      (NOT NOFORCE)
      (NEQ (\TEDIT.PRIMARYW TEXTOBJ)
          (PROCESSPROP (TTY.PROCESS)
                        'WINDOW))
      (TTY.PROCESS (WINDOWPROP (\TEDIT.PRIMARYW TEXTOBJ)
                                'PROCESS)]
(RETURN (fetch (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ))

```

(\TEDIT.WORDDELETE

[LAMBDA (TEXTOBJ)

; Edited 29-May-91 18:22 by jds

;; Delete the word to the left of the caret.

;; Back word.

;; THIS FUNCTION IS FRAUGHT WITH FENCEPOST PROBLEM POTENTIAL, AND THE WHILE vs FOR LOGIC IS CONVOLUTED. CAUTION,
 ;; CAUTION.

```

(LET* ((SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
      (STREAM (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ))
      (READSA (fetch READSA of (OR (fetch (TEXTOBJ TXTWTBL) of TEXTOBJ)
                                   TEDIT.WORDBOUND.READTABLE)))
      (INSCH# (SELECTQ (fetch (SELECTION POINT) of SEL)
                      (LEFT (SUB1 (fetch (SELECTION CH#) of SEL)))
                      (RIGHT (SUB1 (fetch (SELECTION CHLIM) of SEL)))
                      NIL))
      CH CHNO)

```

;; INSCH# is the final (i.e., highest-numbered) character to be deleted.

(COND

```

  ((IGREATERP INSCH# 0)
   (\SETUPGETCH INSCH# TEXTOBJ)
   (SETQ CH (\BIN STREAM))
   (for old CHNO from INSCH# to 1 by -1 while [AND (SELECTC (COND

```

; Don't try to back up past start of file.

```

                                   ((FIXP CH)
                                   (\SYNCODE READSA CH))
                                   (T
                                   ; It's an object!
                                   TEXT.TTC))
                                   (TEXT.TTC NIL)
                                   T)
  (NOT (fetch CLPROTECTED
              of (fetch (PIECE PLOOKS)
                        of (fetch (TEXTSTREAM PIECE)
                                of STREAM]

```

do ;; Skip over any initial separator characters

(SETQ CH (\GETCHB TEXTOBJ)))

;; At this point, CH is the first non-separator character, and CHNO is the character number of the character BEFORE that one.

```

(for old CHNO from CHNO to 1 by -1 while [AND (SELECTC (COND
                                   ((FIXP CH)
                                   (\SYNCODE READSA CH))
                                   (T
                                   ; It's an object!
                                   TEXT.TTC))
                                   (TEXT.TTC T)
                                   NIL)
  (NOT (fetch CLPROTECTED
              of (fetch (PIECE PLOOKS)
                        of (fetch (TEXTSTREAM PIECE) of STREAM]

```

do ;; Skip over the next group of non-separators (= a 'word')

(SETQ CH (\GETCHB TEXTOBJ)))

;; At this point, CH is the first separator character you encountered, and CHNO is the character number of the character BEFORE the
 ;; separator, or 0 if you hit the front of the document.

(\SHOWSEL SEL NIL NIL)

;; First character to delete:

```

[replace (SELECTION CH#) of SEL with (COND
                                   ((ILESSP CHNO 1)
                                   ; Front of document, so start deleting at char # 1
                                   1)
                                   (T
                                   ; Otherwise, we need to start 1 later than the separator we hit,
                                   ; which is 2 higher than CHNO is now.
                                   (IPLUS 2 CHNO)]
(replace (SELECTION CHLIM) of SEL with (ADD1 INSCH#))
(replace (SELECTION DCH) of SEL with (IDIFFERENCE INSCH# CHNO))
(\FIXSEL SEL TEXTOBJ)
(\SHOWSEL SEL NIL T)
(\TEDIT.DELETE SEL TEXTOBJ])

```

(\TEDIT1

[LAMBDA (TEXT WINDOW UNSPAWNED PROPS)

; Edited 12-Jun-90 17:50 by mitani

(* Does the actual editing work, and re-coercion or process kill when done.
Called by TEDIT directly, or ADD.PROCESSED by it.)

```
(SETQ TEXT (OPENTEXTSTREAM TEXT WINDOW NIL NIL PROPS)) (* Open the text for editing)
(\TEDIT.COMMAND.LOOP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)) (* Run the editing engine)
(CLOSEW WINDOW)
(replace (TEXTOBJ \WINDOW) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT) with NIL)
(AND (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
      'AFTERQUITFN)
      (APPLY* (TEXTPROP (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
                      'AFTERQUITFN)
              WINDOW TEXT)) (* Apply any post-window-close (and post-QUIT) function)
(COND
  (UNSPAWNED (* We're not a distinct process%: Send back the edited text in
              some suitable form)
    (COND
      ((NEQ (fetch (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT))
            T)
        (PROG1 (fetch (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT))
          (replace (TEXTOBJ EDITFINISHEDFLG) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT) with NIL)))
      ((STRINGP (fetch (TEXTOBJ TXTFILE) of (fetch (TEXTSTREAM TEXTOBJ) of TEXT)))
        (COERCETEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TEXT)
                       'STRINGP))
      (T TEXT]))
)
```

(MOVD? 'NIL 'OBJECTOUTOFTEDIT)

:: HOOK for looked-string copy, etc. Used in \TEDIT.FOREIGN.COPY?.

```
(DEFINEQ
  (\CREATE.TEDIT.RESTART.MENU
    [LAMBDA NIL
      (CREATE MENU
        ITEMS _ ' (NewEditProcess))
  )
```

:: Added by yabu.fx, for SUNLOADUP without DWIM.

:: Debugging functions

```
(DEFINEQ
  (PLCHAIN
    [LAMBDA (LN)
      (PRINTLINE LN)
      (COND
        ((fetch (LINEDESCRIPTOR NEXTLINE) of LN)
          (PLCHAIN (fetch (LINEDESCRIPTOR NEXTLINE) of LN))
        )
    )
```

; Edited 29-May-91 18:20 by jds

```
(PRINTLINE
  [LAMBDA (LN)
    (printout T "-----" T LN " Bot: " (fetch (LINEDESCRIPTOR YBOT) of LN)
      " Base: "
      (fetch (LINEDESCRIPTOR YBASE) of LN)
      " Height: "
      (fetch (LINEDESCRIPTOR LHEIGHT) of LN)
      " Ascent: "
      (fetch (LINEDESCRIPTOR ASCENT) of LN)
      " Descent: "
      (fetch (LINEDESCRIPTOR DESCENT) of LN)
      T "Char1: " (fetch (LINEDESCRIPTOR CHAR1) of LN)
      " Lim: "
      (fetch (LINEDESCRIPTOR CHARLIM) of LN)
      " Top: "
      (fetch (LINEDESCRIPTOR CHARTOP) of LN))
    (COND
      ((fetch (LINEDESCRIPTOR DIRTY) of LN)
        (PRIN1 " DIRTY"))
      ((fetch (LINEDESCRIPTOR CR\END) of LN)
        (PRIN1 " CR-at-end"))
      ((fetch (LINEDESCRIPTOR DELETED) of LN)
        (PRIN1 " DELETED"))
      ((fetch (LINEDESCRIPTOR LHASPROT) of LN)
        (PRIN1 " [Protected text]"))
      ((fetch (LINEDESCRIPTOR LHASTABS) of LN)
        (PRIN1 " Has Tabs"))
    )
```

; Edited 29-May-91 18:20 by jds

(* Print out a line descriptor in a reasonable form.)

```

(PRIN1 ".
")
(printout T "RMar: " (fetch (LINEDESCRIPTOR RIGHTMARGIN) of LN)
" XLim: "
(fetch (LINEDESCRIPTOR LXLIM) of LN)
" Left: "
(fetch (LINEDESCRIPTOR SPACELEFT) of LN)
T "Prev: " (fetch (LINEDESCRIPTOR PREVLIN) of LN)
T "Next: " (fetch (LINEDESCRIPTOR NEXTLINE) of LN)
T)
(COND
((AND (IGE Q (fetch (LINEDESCRIPTOR CHAR1) of LN)
1)
(ILEQ (fetch (LINEDESCRIPTOR CHAR1) of LN)
(fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))) (* The line is real -- print it.)
(SETUPGETCH (fetch (LINEDESCRIPTOR CHAR1) of LN)
TEXTOBJ)
(PRIN1 " |")
[bind CH for CHNO from (fetch (LINEDESCRIPTOR CHAR1) of LN) to (IMIN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
(fetch (LINEDESCRIPTOR CHARLIM)
of LN))
do (SETQ CH (\GETCH TEXTOBJ))
(COND
((SMALLP CH)
(PRIN1 (CHARACTER CH)))
(T (PRINT CH]
(PRIN1 " |
"]

```

(SEEFILE

```

[LAMBDA (FILE ST ND) (* jds " 4-NOV-83 20:21")
(PROG (CH)
[SETQ FILE (OR (OPENP FILE)
(OPENSTREAM FILE 'INPUT)
(SETFILEPTR FILE (OR ST 0))
(for I from (OR ST 0) to (OR ND (SUB1 (GETEOFPTR FILE)))
do (printout T I 5 (SETQ CH (BIN FILE))
9
(COND
[(ILEQ CH (CHARCODE ^Z))
(CONCAT "^" (CHARACTER (IPLUS CH (CHARCODE @)
(T (CHARACTER CH)))
T])]
)

```

;; Object-oriented editing

(DEFINEQ

(TEDIT.INSERT.OBJECT

```

[LAMBDA (OBJECT STREAM CH#) ; Edited 21-Apr-93 00:52 by jds
;; Inserts the Image-object OBJECT into text STREAM in front of character CH.
(LET* ((TEXTOBJ (TEXTOBJ STREAM))
(SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
SUBSTREAM START-OF-PIECE)
(TEDIT.DO.BLUEPENDINGDELETE SEL TEXTOBJ) ; Do the pending delete, if there is one.
(COND
((NULL CH#) ; Omitted CH# means put it at the current spot.
(SETQ CH# SEL)))
[COND
((type? SELECTION CH#)
;; If the CH# passed in was a selection (or we set it because he defaulted CH#), then compute the REAL CH#.
(SETQ CH# (SELECTQ (fetch (SELECTION POINT) of CH#)
(LEFT (fetch (SELECTION CH#) of CH#))
(RIGHT (fetch (SELECTION CHLIM) of CH#))
(SHOULDNT]
(PROG ((PCTB (ffetch (TEXTOBJ PCTB) of TEXTOBJ))
TEXTLEN PC PCNO CHNO NEWPC PREVPC INSERTFN)
(COND
((fetch (TEXTOBJ TXTREADONLY) of TEXTOBJ)
;; If no changes are allowed to this TEdit, bail out without doing anything.
(RETURN)))
(\SHOWSEL SEL NIL NIL) ; Turn off the selection for now
(SETQ TEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))
(SETQ CH# (IMIN CH# (ADD1 TEXTLEN))) ; CH# we're to insert these characters in front of
(freplace (TEXTOBJ \INSERTFIRSTCH) of TEXTOBJ with -1)
[SETQ PC (COND
((ILEQ CH# TEXTLEN)
(\CHTOPC CH# PCTB T))
(T 'LASTPIECE]
; Piece we're to insert in front of or inside
(SETQ NEWPC (create PIECE
PSTR _ NIL

```

```

        PFILE _ NIL
        POBJ _ OBJECT
        PLEN _ 1)) ; The new piece we're inserting
[COND
  ((SETQ SUBSTREAM (IMAGEOBJPROP OBJECT 'SUBSTREAM))
   ; If this is computed text in bulk, fix the length.
   (replace (PIECE PLEN) of NEWPC with (fetch (TEXTOBJ TEXTLEN) of (fetch (TEXTSTREAM TEXTOBJ)
                                             of SUBSTREAM)))
  (COND
    ((OR (IGREATERP CH# TEXTLEN)
         (IEQP CH# START-OF-PIECE))
     ; We're inserting on a piece boundary; do it, then remember the
     ; prior piece.
     (\INSERTPIECE NEWPC PC TEXTOBJ))
    (T
     ; Not on a piece boundary; split the piece we're inside of, then
     ; insert.
     (\INSERTPIECE NEWPC (\SPLITPIECE PC (IDIFFERENCE CH# START-OF-PIECE)
                                         TEXTOBJ)
                        TEXTOBJ)))
  (COND
    ((SETQ INSERTFN (IMAGEOBJPROP OBJECT 'WHENINSERTEDFN))
     ; If there is a WHENINSERTEDFN, apply it.
     (APPLY* INSERTFN OBJECT (AND (CAR (fetch (TEXTOBJ \WINDOW) of TEXTOBJ))
                                   (WINDOWPROP (CAR (fetch (TEXTOBJ \WINDOW) of TEXTOBJ))
                                                'DSP))
      NIL STREAM)))
  (SETQ PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
  (SETQ PREVPC (fetch (PIECE PREVPiece) of NEWPC)) ; Fill in the para looks
[COND
  [PREVPC (COND
    [(AND (fetch (PIECE PPARALAST) of PREVPC)
          (fetch (PIECE NEXTPIECE) of NEWPC))
     (replace (PIECE PPARALOOKS) of NEWPC with (fetch (PIECE PPARALOOKS)
                                                         of (fetch (PIECE NEXTPIECE) of NEWPC))
     (T (replace (PIECE PPARALOOKS) of NEWPC with (fetch (PIECE PPARALOOKS) of PREVPC))
    (COND
      ((SETQ PREVPC (fetch (PIECE NEXTPIECE) of NEWPC))
       (replace (PIECE PPARALOOKS) of NEWPC with (fetch (PIECE PPARALOOKS) of PREVPC)))
      (T (replace (PIECE PPARALOOKS) of NEWPC with (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
       (replace (PIECE PLOOKS) of NEWPC with (fetch (TEXTOBJ CARETLOOKS) of TEXTOBJ))
      (\TEDIT.HISTORYADD TEXTOBJ (create TEDITHISTORYEVENT
                                         THACTION _ 'Insert
                                         THCH# _ CH#
                                         THLEN _ 1
                                         THFIRSTPIECE _ NEWPC))
      (SETQ TEXTLEN (freplace (TEXTOBJ TEXTLEN) of TEXTOBJ with (IPLUS (fetch (PIECE PLEN) of NEWPC)
                                                                    TEXTLEN)))
      (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL)
      ; Since adding an IMAGEOBJ creates a new piece, the old
      ; insertion cache piece is no longer valid.
      (replace (TEXTOBJ \DIRTY) of TEXTOBJ with T)
      (replace (THISLINE DESC) of (fetch (TEXTOBJ THISLINE) of TEXTOBJ) with NIL)
      (replace (SELECTION SELKIND) of SEL with 'CHAR)
    (COND
      ((fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
       (\FIXILINES TEXTOBJ SEL CH# (fetch (PIECE PLEN) of NEWPC)
        (SUB1 TEXTLEN))
       (replace (SELECTION DCH) of SEL with 0)
       (replace (SELECTION DX) of SEL with 0)
       (TEDIT.UPDATE.SCREEN TEXTOBJ)
       (\FIXSEL SEL TEXTOBJ)
       (\SHOWSEL SEL NIL T))
      (T (replace (SELECTION CHLIM) of SEL with (replace (SELECTION CH#) of SEL
                                                         with (IPLUS CH# (fetch (PIECE PLEN) of NEWPC))
       (replace (SELECTION DCH) of SEL with 0)
       (replace (SELECTION DX) of SEL with 0)
       (replace (SELECTION POINT) of SEL with 'LEFT)
       (replace (THISLINE DESC) of (fetch (TEXTOBJ THISLINE) of TEXTOBJ) with NIL)))
      (\COPYSEL SEL TEDIT.SELECTION])

```

(TEDIT.EDIT.OBJECT

```

  [LAMBDA (STREAM OBJ) ; Edited 29-May-91 18:23 by jds
    (PROG ([TEXTOBJ (COND
      ((type? TEXTOBJ STREAM)
       STREAM)
      ((type? STREAM STREAM)
       (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
      ((SHOULDNT]
      SEL LL CH# SELOBJ EDITFN)
    [COND
      [(AND OBJ (IMAGEOBJP OBJ))
       (SETQ CH# (TEDIT.FIND.OBJECT TEXTOBJ OBJ))
      (COND
        (CH# (SETQ SEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
         (replace (SELECTION CH#) of SEL with CH#)
         (replace (SELECTION CHLIM) of SEL with (ADD1 CH#))
         (SETQ SELOBJ OBJ)

```

```

        (replace (SELECTION DCH) of SEL with 1)
        (replace (SELECTION \TEXTOBJ) of SEL with TEXTOBJ)
        (\FIXSEL SEL TEXTOBJ)
    (T (TEDIT.PROMPTPRINT TEXTOBJ "Can't find specified object." T]
    (T (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
    (SETQ OBJ (fetch (SELECTION SELOBJ) of SEL]
(COND
  [OBJ
    (* OK There's an object selected.
    Edit it.)

    (SETQ EDITFN (IMAGEOBJPROP OBJ 'EDITFN))
    (COND
      ((AND EDITFN (APPLY* EDITFN OBJ))
        (* If the editfn makes a change, update the screen.)
        (for LINE inside (fetch (SELECTION L1) of SEL) do (replace (LINEDESCRIPTOR DIRTY) of LINE
          with T))
        (replace (TEXTOBJ TXTNEEDSUPDATE) of TEXTOBJ with T)
        (TEDIT.UPDATE.SCREEN TEXTOBJ]
      (T
        (* No object selected.)
        (TEDIT.PROMPTPRINT TEXTOBJ "Please select an editable object first." T])

```

(TEDIT.FIND.OBJECT

```

[LAMBDA (TEXTOBJ OBJ)
  (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
  (LET ((PC (\GETBASEPTR (\FIRSTNODE (fetch PCTB of TEXTOBJ)
    0))
    (CH 1))
    (while PC do (COND
      ((AND (NOT (ATOM PC))
        (EQ (fetch (PIECE POBJ) of PC)
          OBJ))
        (RETURN CH))
      (T (add CH (ffetch (PIECE PLEN) of PC))
        (SETQ PC (fetch (PIECE NEXTPIECE) of PC]))
    )
  )
; Edited 3-May-93 12:52 by jds
; Find OBJ, if it's in TEXTOBJ, and return CH#. Else return nil

```

(TEDIT.FIND.OBJECT.SUBTREE

```

[LAMBDA (PCTB OBJ)
  (COND
    ((NULL PCTB)
      NIL)
    ((ATOM (fetch (PCTNODE PCE) of PCTB))
      (OR (TEDIT.FIND.OBJECT.SUBTREE (fetch (PCTNODE LO) of PCTB)
        OBJ)
        (TEDIT.FIND.OBJECT.SUBTREE (fetch (PCTNODE HI) of PCTB)
          OBJ)))
    ((EQ (fetch (PIECE POBJ) of (fetch (PCTNODE PCE) of PCTB)
      OBJ)
      (fetch (PCTNODE CHNUM) of PCTB))
      (T (OR (TEDIT.FIND.OBJECT.SUBTREE (fetch (PCTNODE LO) of PCTB)
        OBJ)
        (TEDIT.FIND.OBJECT.SUBTREE (fetch (PCTNODE HI) of PCTB)
          OBJ])
    )
; Edited 12-Jun-90 17:52 by mitani

```

(TEDIT.PUT.OBJECT

```

[LAMBDA (PIECE OFILE FONTFILE CURCH#)
  (PROG ((OBJECT (fetch (PIECE POBJ) of PIECE))
    (FONTCH# (GETFILEPTR FONTFILE))
    (TOFILE LEN)
    (\DWOUT FONTFILE 0)
    (\SMALLPOUT FONTFILE \PieceDescriptorOBJECT)
    (\ATMOUT FONTFILE (IMAGEOBJPROP OBJECT 'GETFN))
    (APPLY* (IMAGEOBJPROP OBJECT 'PUTFN)
      OBJECT OFILE)
    (SETFILEPTR FONTFILE FONTCH#)

    (* Now go back and fill in the length of the text description of the object.)

    [\DWOUT FONTFILE (SETQ LEN (ADD1 (IDIFFERENCE (GETEOFPTR OFILE)
      CURCH#])
    (SETFILEPTR FONTFILE -1)
    (AND (RANDACCESSP OFILE)
      (SETFILEPTR OFILE -1))
    (RETURN LEN])
  )
; Edited 12-Jun-90 17:49 by mitani
(* Given a piece which describes an object, put the object out
there.)
(* Placeholder for length of the object's description)
(* Mark this as setting the piece's looks)
(* The FN to apply to reconstruct the object)
(* Make sure we're at the end of the font file)
(* And the text part of the file)

```

(TEDIT.GET.OBJECT

```

[LAMBDA (STREAM PIECE FILE CURCH#)
  (PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
    (FILEPTRSAVE NAMELEN GETFN OBJ NBYTES)
  )
; Edited 12-Jun-90 17:50 by mitani
; Get an object from the file
; CURCH# = fileptr within the text section of the file where the
; object's text starts.

;; rrb 10-AUG-87 --- calculate the length of the image object's data. This assumes that the file is currently pointed at the end of the data which is

```

;; where the GETFN is written {I think}

```

(SETQ NBYTES (DIFFERENCE (GETFILEPTR FILE)
                          CURCH#))
(SETQ GETFN (\ATMIN FILE))
(SETQ FILEPTRSAVE (GETFILEPTR FILE))
(SETFILEPTR FILE CURCH#)
(SETQ OBJ (READIMAGEOBJ FILE GETFN NIL NBYTES))
(COND
  ((IMAGEOBJPROP OBJ 'UNKNOWNGETFN)
   ; If the object has an unknown getfn property, then it's an
   ; encapsulated object. Warn the user
   (TEDIT.PROMPTPRINT STREAM "WARNING: Document contains unknown image objects." T))
  (SETFILEPTR FILE FILEPTRSAVE)
  (replace (PIECE POBJ) of PIECE with OBJ)
  (replace (PIECE PFILE) of PIECE with NIL)
  (replace (PIECE PSTR) of PIECE with NIL)
  [replace (PIECE PLOOKS) of PIECE with (COND
    ((fetch (PIECE PREVPIECE) of PIECE)
     (fetch (PIECE PLOOKS) of (fetch (PIECE PREVPIECE) of PIECE)))
    (T (OR (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
            (\TEDIT.UNIQUIFY.CHARLOOKS (CHARLOOKS.FROM.FONT
                                         DEFAULTFONT)
                                         TEXTOBJ]
    (RETURN (fetch (PIECE POBJ) of PIECE))

```

(TEDIT.OBJECT.CHANGED

[LAMBDA (STREAM OBJECT)

; Edited 12-Jun-90 17:51 by mitani

(* Notify TEdit that an object has changed, and the display may need to be updated.)

```

(PROG ((TEXTOBJ (TEXTOBJ STREAM))
  (LINES (fetch (TEXTOBJ LINES) of (TEXTOBJ STREAM)))
  PCINFO CHANGED CHANGEDCH#)
(SETQ PCINFO (TEDIT.MAPPICES TEXTOBJ [FUNCTION (LAMBDA (CH# PC PCNO OBJ)
  (AND (EQ OBJ (fetch (PIECE POBJ) of PC))
    'STOP]
  (* Find the piece containing this object)
  (OR PCINFO (HELP "Changed OBJECT not found!?"))
  (* Get the CH# of the changed object)
  (SETQ CHANGEDCH# (CAR PCINFO))
  (\TEDIT.MARK.LINES.DIRTY TEXTOBJ CHANGEDCH# CHANGEDCH#)
  (* Mark affected lines)
  (replace (TEXTOBJ \DIRTY) of TEXTOBJ with T)
  (* And mark the document dirty.)
  (\SHOWSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
    NIL NIL)
  (TEDIT.UPDATE.SCREEN TEXTOBJ)
  (\FIXSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
    TEXTOBJ)
  (\SHOWSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
    NIL T])
)

```

```

(FILESLoad TEDITFIND TEDITHistory TEDITFILE TEDITWINDOW TEDITSELECTION IMAGEOBJ TFBRAVO TEDITHCpy TEDITPAGE
TEDITMENU TEDITFNKEYS)

```

;; TEDIT Support information

(RPAQQ TEDITSYSTEMDATE " 4-May-95 10:37:23")

(RPAQ TEDITSupport "TEditSupport.PA")

(DEFINEQ

(MAKETEDITFORM

[LAMBDA NIL

(* jds "12-Mar-85 04:00")

(* Builds a trouble-report form for TEdit.)

(MAKEXXSUPPORTFORM "TEdit" TEDITSupport TEDITSYSTEMDATE])

)

(ADDTovar LAFITESPECIALFORMS ("TEdit Report" 'MAKETEDITFORM "Report a problem with TEdit"))

(SETQ LAFITEFORMSMENU NIL)

;; LISTFILES Interface, so the system can decide if a file is a TEdit file.

```

(ADDTovar PRINTFILETYPES (TEDIT (TEST \TEDIT.FORMATTEDP1)
  (EXTENSION (TEDIT))))

```

```

(PUTPROPS TEDIT COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993
  1995))

```

FUNCTION INDEX

COERCETEXTOBJ	2	TEDIT.INSERT	6	\TEDIT.CHARDELETE	13
MAKETEDITFORM	22	TEDIT.INSERT.OBJECT	19	\TEDIT.COPY.PIECEMAPFN	14
PLCHAIN	18	TEDIT.KILL	7	\TEDIT.DELETE	14
PRINTLINE	18	TEDIT.MAPLINES	7	\TEDIT.DIFFUSE.PARALOOKS	15
SEEFIL	19	TEDIT.MAPPICES	7	\TEDIT.FOREIGN.COPY?	16
TEDIT	3	TEDIT.MOVE	7	\TEDIT.INCLUDE	10
TEDIT.CHARWIDTH	4	TEDIT.OBJECT.CHANGED	22	\TEDIT.INSERT.PIECES	11
TEDIT.COPY	4	TEDIT.PUT.OBJECT	21	\TEDIT.MOVE.PIECEMAPFN	12
TEDIT.DELETE	5	TEDIT.QUIT	9	\TEDIT.OBJECT.SHOWSEL	12
TEDIT.DO.BLUEPENDINGDELETE	5	TEDIT.STRINGWIDTH	9	\TEDIT.QUIT	16
TEDIT.EDIT.OBJECT	20	TEDIT.\INSERT	9	\TEDIT.RESTARTFN	13
TEDIT.FIND.OBJECT	21	TEXTOBJ	10	\TEDIT.WORDDELETE	17
TEDIT.FIND.OBJECT.SUBTREE	21	TEXTSTREAM	10	\TEDIT1	17
TEDIT.GET.OBJECT	21	\CREATE.TEDIT.RESTART.MENU	18	\TEDIT2	2

VARIABLE INDEX

LAFITESPECIALFORMS	22	TEDIT.RESTART.MENU	1	TEDIT.TERMSA.FONT	1
PRINTFILETYPES	22	TEDIT.STARTUP.MONITORLOCK	1	TEDIT.SUPPORT	22
TEDIT.DEFAULT.PROPS	1	TEDIT.TENTATIVE	1	TEDIT.SYSTEMDATE	22

CONSTANT INDEX

\SCRATCHLEN	1
-------------------	---
