

File created: 4-May-95 10:30:33 {DSK}<lispcore>library>new>TEDITFILE.;2

changes to: (FNS TEDIT.INCLUDE TEDIT.BUILD.PCTB0)

previous date: 25-Aug-94 10:53:27 {DSK}<king>export>lispcore>library>TEDITFILE.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ TEDITFILECOMS
  ((FILES TEDITDCL)
   (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (\SCRATCHLEN 64))
    (FILES (LOADCOMP)
            TEDITDCL)))
  (COMS ;; GETting a file
    (FNS TEDIT.BUILD.PCTB \TEDIT.CONVERT.FOREIGN.FORMAT TEDIT.FORMATTEDFILEP TEDIT.GET
      TEDIT.PARSE.PAGEFRAMES1 \ARBIN \ATMIN \DWIN \STRINGIN \TEDIT.FORMATTEDP1 \TEDIT.SET.WINDOW))
  (COMS ;; INCLUDEing a file
    (FNS TEDIT.INCLUDE TEDIT.RAW.INCLUDE))
  (COMS ;; PUTting a file:
    (FNS TEDIT.PUT TEDIT.PUT.PCTB \TEDIT.PUTRESET TEDIT.PUT.PIECE.DEScriptor \ARBOU \ATMOUT \DWOUT
      \STRINGOUT \TEDIT-OPEN-FONT-FILE))
  (FNS \TEDIT.GET.CHARLOOKS.LIST \TEDIT.GET.SINGLE.CHARLOOKS \TEDIT.PUT.CHARLOOKS.LIST
    \TEDIT.PUT.SINGLE.CHARLOOKS)
  (FNS \TEDIT.GET.PARALOOKS.LIST \TEDIT.GET.SINGLE.PARALOOKS \TEDIT.PUT.PARALOOKS.LIST
    \TEDIT.PUT.SINGLE.PARALOOKS)
  (GLOBALVARS TEDIT.INPUT.FORMATS *TEDIT-FILE-READTABLE*)
  (INITVARS (TEDIT.INPUT.FORMATS NIL)
    (*TEDIT-FILE-READTABLE* (COPYREADTABLE \ORIGREADTABLE)))
    ; For consistent reading and writing of info on TEdit files.
  )
  (COMS ;; For converting old incoming format. Cutover 5/22/85 to permit looks changes in the future.
    (FNS TEDIT.BUILD.PCTB2 \TEDIT.GET.CHARLOOKS.LIST2 \TEDIT.GET.SINGLE.CHARLOOKS2
      \TEDIT.PUT.SINGLE.PARALOOKS2 \TEDIT.PUT.SINGLE.CHARLOOKS2 \TEDIT.GET.PARALOOKS.LIST2
      \TEDIT.GET.SINGLE.PARALOOKS2 TEDIT.PUT.PCTB2 \TEDIT.PUT.CHARLOOKS.LIST2
      \TEDIT.PUT.PARALOOKS.LIST2))
  (COMS ;; For converting incoming old-format files (1/27/85 cutover)
    (FNS TEDIT.BUILD.PCTB1 TEDIT.GET.PAGEFRAMES1 \TEDIT.GET.CHARLOOKS1 \TEDIT.GET.PARALOOKS1
      TEDIT.GET.OBJECT1))
  (COMS ;; VERSION 0 Compatibility reading functions
    (FNS TEDIT.BUILD.PCTB0 TEDIT.GET.CHARLOOKS0 TEDIT.GET.OBJECT0 TEDIT.GET.PARALOOKS0))))

(FILESLoad TEDITDCL)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ \SCRATCHLEN 64)

(CONSTANTS (\SCRATCHLEN 64))
)

(FILESLoad (LOADCOMP)
  TEDITDCL)
)

;; GETting a file

(DEFINEQ

(TEDIT.BUILD.PCTB
  [LAMBDA (TEXT TEXTOBJ START END DEFAULTLOOKS DEFAULTPARALOOKS CLEARGET?)
    ; Edited 19-Apr-93 13:46 by jds
    ; START = 1st char of file to read from, if specified
    ; END = use this as eofptr of file. For use in reading files within
    ; files.
    (PROG (SEL LINES PCTB PC OLDPC PCCOUNT TYPECODE PCLEN CHLOOKSSEEN NEWPC PARALOOKSSEEN PIECEINFOCH# CACHE
      CACHE? TTEXTOBJ USER.CMFILE TSTREAM USERFILEFORMAT USERTEMP EXISTINGCHARLOOKS EXLOOK
      EXISTINGFMTSPECS (*READTABLE* *TEDIT-FILE-READTABLE*)
      (*PRINT-BASE* 10)
      (CURFILECH# (OR START 0))
      (CURCH# 1)
      (TEXTSTREAM (AND TEXTOBJ (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ))))
```

```

LOOKSHASH PARAHASH)
[SETQ DEFAULTPARALOOKS (OR DEFAULTPARALOOKS (COND
    (TEXTOBJ (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
    (T (create FMTSPEC using TEDIT.DEFAULT.FMTSPEC]
        ; Set the default paragraph formatting for filling in piece
        ; PPARALOOKS fields
(COND
    (TEXTOBJ
        ; If there's a TEXTOBJ behind this, set its TXTFILE field to point
        ; to the right place.
        (replace (TEXTOBJ TXTFILE) of TEXTOBJ with TEXT)))
(SETQ DEFAULTLOOKS (OR DEFAULTLOOKS (CHARLOOKS.FROM.FONT DEFAULTFONT)))
    ; Set the default CHARLOOKS, for filling in pieces' PLOOKS
    ; fields
(SETQ TEXT (\CREATEPIECEORSTREAM TEXT DEFAULTLOOKS DEFAULTPARALOOKS START END))
    ; Grab the file, or a single piece (if the text is a string, or such
    ; simple cases)
(AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with NIL))
    ; Start by assuming no page formatting
(COND
    ((STREAMP TEXT)
        ; OK, it wasn't a string, so check for cases where we have to
        ; cache the file locally.
        (AND TEXTOBJ (replace (TEXTOBJ TXTFILE) of TEXTOBJ with TEXT))
        (COND
            ((OR [AND TEXTOBJ (SETQ CACHE? (TEXTPROP TEXTOBJ 'CACHE)
                (NOT (RANDACCESSP TEXT)))
                ; If the file device isn't random access, cache the file locally.
                ; Also do this if he asks for a local cache.
                (SETQ CACHE (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW))
                ; The cache file
                (COND
                    ((OR START END)
                     (COPYBYTES TEXT CACHE (OR START 0)
                                     (OR END -1)))
                     (T (COPYBYTES TEXT CACHE)))
                    ; Copy the text there
                    (SETQ CACHE? T)
                    ; Remember that we cached it!
                    ;; COPYBYTES can only have start/end args of NIL if the file is not random access. So it's impossible to grab out of the middle
                    ;; of a file on an NS server. Sorry.
                    (COND
                        (CACHE?
                         ;; for the folx who don't trust the connections, since all their pcs will point to core, we can close the txtfile
                         ;; connection
                         (CLOSEF TEXT)))
                    (replace (STREAM EOLCONVENTION) of CACHE with (fetch (STREAM EOLCONVENTION) of TEXT))
                    ; Remember the EOL convention from the original file, so that we
                    ; can do a copybytes if need by.
                    ; And pretend the cache IS the real file from here on
                    (SETQ TEXT CACHE)
                    (SETQ START (SETQ END NIL))
                    ;; Since we only copied the relevant part of the file into the cache, we don't need to remember the limits of interest.
                    ))
            (SETQ PCCOUNT (\TEDIT.FORMATTEDP1 TEXT END))
            (COND
                ((AND (NOT PCCOUNT)
                     (NEQ (fetch (STREAM EOLCONVENTION) of TEXT)
                          CR.EOLC))
                 ;; This is an UNFORMATTED file, and it has a foreign EOL convention. Convert it, and save the converted copy locally.
                 (SETQ CACHE (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW))
                 ; Build a cache file
                 (COND
                     ((OR START END)
                      (COPYCHARS TEXT CACHE (OR START 0)
                                      (OR END -1))
                      ;; mcf: just like before, we have all the relevant portion
                      (SETQ START (SETQ END NIL)))
                     (T (COPYCHARS TEXT CACHE)))
                     ; Copy the text, converting from the foreign EOL convention into
                     ; CR as end of line.
                     (SETQ TEXT CACHE)
                     ;; And think of THIS as the cache. At this point, we may have cached twice in succession--no need to clip off START and END.
                     (SETQ CACHE? T)
                     ; Remember that we cached the file!
                     ))
                ))
            ; Check to see if this is a formatted file, and find out how may
            ; pieces we should allocate for it.
            ))
        (AND TEXTOBJ (TEXTPROP TEXTOBJ 'CACHE CACHE?))
        ; REMEMBER THAT THIS TEXT WAS CACHED, SO THAT
        ; LATER PUTS DON'T INVALIDATE THE CACHE.
        (COND
            [(type? PIECE TEXT)
                ; If this isn't a text stream, build a piece table with the one piece
                ; in it.
                (COND
                    ((EQ (fetch (PIECE PLEN) of TEXT)
                        0)
                     ; I hate piece whose length is zero.
                     (* INSERT-BRT (CREATEPCNODE 1

```

```

)
(T (SETQ PCTB (\MAKEPCTB TEXT))
    (QUOTE LASTPIECE)) PCTB)
(* INSERT-BRT (CREATEPCNODE
  (ADD1 (fetch (PIECE PLEN) of TEXT))
  (QUOTE LASTPIECE)) PCTB)
(replace (PIECE PLOOKS) of TEXT with (\TEDIT.UNIQUIFY.CHARLOOKS (fetch (PIECE PLOOKS)
  of TEXT)
  TEXTOBJ))
; And note the CHARLOOKS and PARALOOKS of this text--as
; well as filling them in.
(replace (PIECE PPARALOOKS) of TEXT with (\TEDIT.UNIQUIFY.PARALOOKS (fetch (PIECE PPARALOOKS)
  of TEXT)
  TEXTOBJ)
(CLEARGET?
;; If the user wants an uninterpreted stream onto the file , build a piece table with the one piece in it.
(SETQ TEXT (create PIECE
  PFILE _ TEXT
  PFPOS _ (COND
    (START START)
    (T 0))
  PLEN _ (IDIFFERENCE (OR END (GETEOFPTR TEXT))
    (COND
      (START START)
      (T 0)))
  PREVPIECE _ NIL
  PLOOKS _ DEFAULTLOOKS
  PPARALAST _ NIL
  PPARALOOKS _ DEFAULTTPARALOOKS))
; A single piece to describe the whole file
(SETQ PCTB (\MAKEPCTB TEXT))
(replace (PIECE PLOOKS) of TEXT with (\TEDIT.UNIQUIFY.CHARLOOKS (fetch (PIECE PLOOKS)
  of TEXT)
  TEXTOBJ))
; And note the CHARLOOKS and PARALOOKS for later saving.
; Keep those caches consistent.
(replace (PIECE PPARALOOKS) of TEXT with (\TEDIT.UNIQUIFY.PARALOOKS (fetch (PIECE PPARALOOKS)
  of TEXT)
  TEXTOBJ))
(* INSERT-BRT (CREATEPCNODE
  (ADD1 (fetch (PIECE PLEN) of TEXT))
  (QUOTE LASTPIECE)) PCTB)
)
[ (NOT PCCOUNT)
  (COND
    [(SETQ USERFILEFORMAT (for FILETYPE in TEDIT.INPUT.FORMATS when (SETQ USERTEMP
      (APPLY* (CAR FILETYPE)
        TEXT))
      do (RETURN FILETYPE))) ; The input file is in a user-sensible format, which he is willing to
      ; convert for TEdit's use.
      ; See if there are Bravo headers
    (SETQ PCTB (\TEDIT.CONVERT.FOREIGN.FORMAT (CADR USERFILEFORMAT)
      TEXT USERTEMP TEXTOBJ DEFAULTLOOKS DEFAULTTPARALOOKS))
      ; Convert the foreign format file, and grab its PCTB
    (bind (PC _ (\GETBASEPTR (\FIRSTNODE PCTB)
      0))
      while [AND PC (NOT (EQ PC 'LASTPIECE)] do
        ; Run thru the converted pieces, noting their CHARLOOKS and
        ; PARALOOKS for the get/put caches.
        (replace (PIECE PLOOKS) of PC
          with (\TEDIT.UNIQUIFY.CHARLOOKS
            (fetch (PIECE PLOOKS) of PC)
            TEXTOBJ))
        (replace (PIECE PPARALOOKS) of PC
          with (\TEDIT.UNIQUIFY.PARALOOKS
            (fetch (PIECE PPARALOOKS) of PC)
            TEXTOBJ))
        (SETQ PC (fetch (PIECE NEXTPIECE) of PC]
          ; Nope--it's straight unformatted text
    (T
      [SETQ PCTB (\MAKEPCTB (create PIECE
        PFILE _ TEXT
        PFPOS _ CURFILECH#
        PLEN _ (IDIFFERENCE (OR END (GETEOFPTR TEXT))
          CURFILECH#)
        PREVPIECE _ NIL
        PLOOKS _ (\TEDIT.UNIQUIFY.CHARLOOKS DEFAULTLOOKS TEXTOBJ)
        PPARALAST _ NIL
        PPARALOOKS _ (\TEDIT.UNIQUIFY.PARALOOKS DEFAULTTPARALOOKS
          TEXTOBJ)
        ; So create a single piece to describe its contents
        (* INSERT-BRT (CREATEPCNODE
          (ADD1 (IDIFFERENCE (OR END
            (GETEOFPTR TEXT)) CURFILECH#))
          (QUOTE LASTPIECE)) PCTB)
        ; Insert LASTPIECE here
      ]
    ] (LISTP PCCOUNT)
      ; This is an obsolete version of the TEdit file format.

```

```

(SELECTQ (CAR PCCOUNT)
  (0
    (SETQ PCTB (TEDIT.BUILD.PCTB0 TEXT TEXTOBJ (CDR PCCOUNT)
      START END)))
    ; VERSION 0
    ; PARALOOKS for the caches.
  (1
    (SETQ PCTB (TEDIT.BUILD.PCTB1 TEXT TEXTOBJ (CDR PCCOUNT)
      START END)))
    ; Version 1; obsolete at INTERMEZZO release 2/85
  (2
    (SETQ PCTB (TEDIT.BUILD.PCTB2 TEXT TEXTOBJ (CDR PCCOUNT)
      START END)))
    ; Version 2; obsolete 5/22/85
  (SHOULDNT "File format version incompatible with this version of TEdit."))
(bind (PC _ (\GETBASEPTR (\FIRSTNODE PCTB)
  0))
  while [AND PC (NOT (EQ PC 'LASTPIECE] do
    ; Run thru the converted pieces, noting CHARLOOKS and
    ; PARALOOKS for the caches.
    (replace (PIECE PLOOKS) of PC
      with (\TEDIT.UNIQUIFY.CHARLOOKS (fetch (PIECE PLOOKS)
        of PC)
        TEXTOBJ))
    (replace (PIECE PPARALOOKS) of PC
      with (\TEDIT.UNIQUIFY.PARALOOKS (fetch (PIECE
        PPARALOOKS
        )
        of PC)
        TEXTOBJ))
    (SETQ PC (fetch (PIECE NEXTPIECE) of PC]
    ; This IS a TEdit-format file, so read in all the parts.
  (T
    (SETQ PCTB (\MAKEPCTB NIL PCCOUNT))
    (replace (TEXTOBJ PCTB) of TEXTOBJ with PCTB)
    (SETFILEPTR TEXT (IDIFFERENCE (OR END (GETEOFPTR TEXT))
      8))
    (SETQ PIECEINFOCH# (\DWIN TEXT))
    (SETFILEPTR TEXT PIECEINFOCH#)
    (bind (OLDPC _ NIL)
      (OLDPARALOOKS _ DEFAULTPARALOOKS) for I from 1 to PCCOUNT as PCN from 1
      do (SETQ PC NIL)
        ; This loop may not really read a piece, so we have to distinguish
        ; that case.
        (SETQ PCLEN (\DWIN TEXT))
        (SETQ TYPECODE (\SMALLPIN TEXT))
        ; What kind of piece is it?
        [SELECTC TYPECODE
          (\PieceDescriptorPAGEFRAME
            (AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with (TEDIT.GET.PAGEFRAMES
              TEXT)))
            (add PCN -1)
            ;; (|add| pcn (|minus| |\EltsPerPiece|))
            ; This didn't create a piece -- don't count it in the PCTB
            ; placement.
          )
          (\PieceDescriptorCHARLOOKSLIST
            (replace (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ with (\TEDIT.GET.CHARLOOKS.LIST TEXT
              TEXTOBJ))
            ; Read the list of looks used in this document.
            [SETQ LOOKSHASH (ARRAY (FLENGTH (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ]
              ; Build an array of the looks, so the reader can index them.
            (for I from 1 as LOOKS in (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ)
              do (SETA LOOKSHASH I LOOKS))
            (add PCN -1)
            ;; (|add| pcn (|minus| |\EltsPerPiece|))
            ; This didn't create a piece -- don't count it in the PCTB
            ; placement.
            (add I -1))
          (\PieceDescriptorPARALOOKSLIST
            ; This is the list of PARALOOKSs used in this document.
            (replace (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ with (\TEDIT.GET.PARALOOKS.LIST TEXT
              TEXTOBJ))
            ; Read the list of looks used in this document.
            [SETQ PARASHASH (ARRAY (FLENGTH (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ]
              ; Build an array of the looks, so the reader can index them.
            (for I from 1 as LOOKS in (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ)
              do (SETA PARASHASH I LOOKS))
            (add PCN -1)
            ;; (|add| pcn (|minus| |\EltsPerPiece|))
            ; This didn't create a piece -- don't count it in the PCTB
            ; placement.
            (add I -1))
          (\PieceDescriptorPARA
            ; Reading a new set of paragraph looks.
            (AND OLDPC (replace (PIECE PPARALAST) of OLDPC with T))
            ; Mark the end of the preceding paragraph.
            (SETQ OLDPARALOOKS (\TEDIT.GET.PARALOOKS TEXT PARASHASH))
            ; Get the new set of looks, for use by later pieces.
            (AND TEXTOBJ (replace (TEXTOBJ FORMATTEDP) of TEXTOBJ with T))
            ; Mark the document as containing paragraph formatting info
            (add PCN -1)
            ;; (|add| pcn (|minus| |\EltsPerPiece|))
            ; This didn't create a piece -- don't count it in the PCTB
            ; placement.
          )
          (\PieceDescriptorLOOKS
            ; New character looks. Build a piece to describe those
            ; characters.

```

```

      (SETQ PC
        (create PIECE
          PFILE _ TEXT
          PFPOS _ CURFILECH#
          PLEN _ PCLEN
          PREVPiece _ OLDPC
          PPARALOOKS _ OLDPARALOOKS))
          ; Build the new piece
      (\TEDIT.GET.CHARLOOKS PC TEXT LOOKSHASH OLDPC)
          ; Read the character looks for this guy.
      (COND
        (OLDPC
          (replace (PIECE NEXTPIECE) of OLDPC with PC)))
          ; If there's a prior piece, hook this one on the chain.
      (add CURFILECH# PCLEN)
          ; And note the passing of characters.
    )
    (\PieceDescriptorOBJECT
      ; It describes an object. Read that, and perhaps some
      ; description of the charlooks to go with it.
      (SETQ PC
        (create PIECE
          PFILE _ TEXT
          PFPOS _ CURFILECH#
          PLEN _ PCLEN
          PREVPiece _ OLDPC
          PPARALOOKS _ OLDPARALOOKS))
      (COND
        (OLDPC
          (replace (PIECE NEXTPIECE) of OLDPC with PC)))
          ; If there's a prior piece, hook this one on the chain.
      (TEDIT.GET.OBJECT TEXTSTREAM PC TEXT CURFILECH#)
      (add CURFILECH# (fetch (PIECE PLEN) of PC))
      [COND
        ((NOT (ZEROP (\BIN TEXT)))
          (replace (PIECE PLOOKS) of PC with (\TEDIT.GET.SINGLE.CHARLOOKS TEXT)))
          ; There are new character looks for this object. Read them in.
        (T
          (replace (PIECE PLOOKS) of PC with (OR (AND OLDPC (fetch (PIECE PLOOKS)
            of OLDPC))
            DEFAULTLOOKS])
          ; No new looks; steal them from the prior piece.
          (replace (PIECE PLOOKS) of PC with (OR (AND OLDPC (fetch (PIECE PLOOKS)
            of OLDPC))
            DEFAULTLOOKS])
          ; OBJECTs are officially one character long.
        )
      (PROGN (TEDIT.PROMPTPRINT TEXTOBJ "WARNING: Unknown-type piece skipped." T)
        (SETFILEPTR TEXT (IPLUS (GETFILEPTR TEXT)
          (\SMALLPIN TEXT)
        )
      )
      (COND
        (PC (\INSERTPIECE PC 'LASTPIECE TEXTOBJ)
          ; If we created a piece, save it in the table.
          (add CURCH# (fetch (PIECE PLEN) of PC))
          (SETQ OLDPC PC)))
      finally
        ;; (\editseta pctx pcn curch#)
        ;; (\editseta pctx (add1 pcn) 'lastpiece)
        ;; (\editseta pctx |\PCTBLastPieceOffset| (add1 pcn))
        ;; (\editseta pctx |\PCTBFreePieces| 0)
        (* INSERT-BRT (CREATEPCNODE CURCH#
          (QUOTE LASTPIECE)) PCTB)
      ]
      (AND (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
        (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
          TEXTOBJ))
          ; And make sure that the default and caret looks are reflected in
          ; that list.
      (AND (fetch (TEXTOBJ CARETLOOKS) of TEXTOBJ)
        (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ CARETLOOKS) of TEXTOBJ)
          TEXTOBJ))
      (AND DEFAULTLOOKS (\TEDIT.UNIQIFY.CHARLOOKS DEFAULTLOOKS TEXTOBJ))
          ; And the default looks we used in this function...
      (\TEDIT.UNIQIFY.PARALOOKS (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)
        TEXTOBJ)
          ; And make sure the default paraloooks are reflected in that list.
      [AND TEXT (bind (CHARLOOKSLIST _ (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ))
        (PARALOOKSLIST _ (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
        for (PC _ (\GETBASEPTR (\FIRSTNODE PCTB)
          0))
        by (fetch (PIECE NEXTPIECE) of PC) while [AND PC (NOT (EQ PC 'LASTPIECE))
        do
          ; Look at every piece, and assure that its CHARLOOKS and
          ; PARALOOKS are in the cache.
          [COND
            ((FMEMB (fetch (PIECE PLOOKS) of PC)
              CHARLOOKSLIST)
              ; This piece's CHARLOOKS are known in the cache already.
              ; Don't bother doing anything else.
            )
            (T
              (replace (PIECE PLOOKS) of PC with (\TEDIT.UNIQIFY.CHARLOOKS (fetch (PIECE PLOOKS)
                of PC)
                TEXTOBJ])
              ; Nope; add these looks to the cache
            )
          ]
          (COND
            ((FMEMB (fetch (PIECE PPARALOOKS) of PC)
              PARALOOKSLIST)
              ; This piece's PARALOOKS are known in the cache already.
            )
          ]

```

; Don't bother doing anything else.

```

)
(T
  (replace (PIECE PPARALOOKS) of PC with (\TEDIT.UNIQUIFY.PARALOOKS
    (fetch (PIECE PPARALOOKS) of PC)
    TEXTOBJ])
  (RETURN PCTB])

```

(\TEDIT.CONVERT.FOREIGN.FORMAT

```

[LAMBDA (CONVERSIONFN FILE PREDICATERESULT TEXTOBJ DEFAULTLOOKS DEFAULTPPARALOOKS)
; Edited 12-Jun-90 18:16 by mitani

```

(* Perform the conversion from a foreign file format into TEdit-internal form as an open TextStream.)

```

(PROG (TSTREAM TTEXTOBJ SEL WORKINGSTREAM) (* See if there are Bravo headers)
  (SETQ WORKINGSTREAM (OPENTEXTSTREAM ""))
  (RESETLST
    (RESETSAVE (\TEDIT.SET.WINDOW (CONS (TEXTOBJ WORKINGSTREAM)
      NIL)))
    (SETQ TSTREAM (APPLY* CONVERSIONFN FILE PREDICATERESULT WORKINGSTREAM)))
  (COND
    (TEXTOBJ

```

(* If we're filling in an existing TEXTOBJ, there are fields that need to be copied.)

```

    (OR (fetch (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ)
      (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with (fetch (TEXTOBJ TXTPAGEFRAMES)
        of (TEXTOBJ TSTREAM))
      (* Such as the page formatting, which the converter may well
        set.)
    ))
  (RETURN (fetch (TEXTOBJ PCTB) of (TEXTOBJ TSTREAM]))

```

(TEDIT.FORMATTEDFILEP

```

[LAMBDA (STREAM)
; Edited 19-Apr-93 11:57 by jds
; Test to see if this stream's text would need a TEdit-format file
; (T) or is just plain text (NIL)

```

```

(PROG ((TEXTOBJ (TEXTOBJ STREAM))
  (FONTFILE 0)
  OLDPPARALOOKS PC OLDLOOKS PREVPC TENTATIVE)
  (SETQ OLDPPARALOOKS (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
  (SETQ TENTATIVE (TEXTPROP TEXTOBJ 'TEDIT.TENTATIVE)) ; If edits are to be shown
  (SETQ PC (\GETBASEPTR (\FIRSTNODE (fetch (TEXTOBJ PCTB) of TEXTOBJ))
    0)) ; First piece in the document
  (COND
    ((ATOM PC) ; Empty document
      (RETURN NIL)))
  (SETQ OLDLOOKS (OR (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
    TEDIT.DEFAULT.CHARLOOKS))
  (while PC do [COND
    ((fetch (PIECE POBJ) of PC) ; OBJECTS require the special format
      (SETQ FONTFILE 4))
    ((AND (OR (NOT PREVPC)
      (fetch (PIECE PPARALAST) of PREVPC))
      (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
        (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)))
      ; We just hit a paragraph break.
      (SETQ FONTFILE (IMAX FONTFILE 3)))
    ([OR (NOT (EQCLOOKS OLDLOOKS (fetch (PIECE PLOOKS) of PC)))
      (AND TENTATIVE (OR (AND PREVPC (NEQ (fetch (PIECE PNEW) of PREVPC)
        (fetch (PIECE PNEW) of PC)))
        (AND (NOT PREVPC)
          (fetch (PIECE PNEW) of PC))
        (AND PREVPC (NEQ (fetch (PIECE PFATP) of PREVPC)
          (fetch (PIECE PFATP) of PC))
          ; Change in font, size, etc.
          (SETQ FONTFILE (IMAX FONTFILE 2)))
      ((fetch (PIECE PFATP) of PC) ; NS Chars in the piece.
        (SETQ FONTFILE (IMAX FONTFILE 1]
      (SETQ PREVPC PC)
      (SETQ PC (fetch (PIECE NEXTPIECE) of PC)))
  (RETURN (SELECTQ FONTFILE
    (0 NIL)
    (1 'NSCHARS)
    (2 'CHARLOOKS)
    (3 'PARALOOKS)
    (4 'IMAGEOBJ)
    NIL])

```

(TEDIT.GET

```

[LAMBDA (TEXTOBJ FILE UNFORMATTED?)
; Edited 19-Apr-93 13:12 by jds

```

;; Get a new file (overwriting the one being edited.)

```

(PROG ((TEXTOBJ (TEXTOBJ TEXTOBJ))
  (OFILE OCURSOR LINES USER.CMFILE RESP TITLE FILENAME MENUSTREAM (GETFN (TEXTPROP TEXTOBJ 'GETFN))
  (SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
  (PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
  (TEDIT.GET.FINISHEDFORMS NIL))
(COND
  ([AND (fetch (TEXTOBJ \DIRTY) of TEXTOBJ)
    (PROGN (AND (fetch (TEXTOBJ PROMPTWINDOW) of TEXTOBJ)
      (FRESHLINE (fetch (TEXTOBJ PROMPTWINDOW) of TEXTOBJ)))
    (NOT (MOUSECONFIRM "Not saved yet; LEFT go Get anyway." T (fetch (TEXTOBJ PROMPTWINDOW)
      of TEXTOBJ])

  ;; Only do the GET if he knows he'll zorch himself.
  (RETURN)))
[SETQ OFILE (OR FILE (\TEDIT.MAKEFILENAME (TEDIT.GETINPUT TEXTOBJ "File to GET: "
  (OR (TEXTPROP TEXTOBJ 'LASTGETFILENAME)
  (\TEXTSTREAM.FILENAME TEXTOBJ]
(TEXTPROP TEXTOBJ 'LASTGETFILENAME OFILE)
(COND
  [(AND OFILE (OR (OPENP FILE)
    (INFILE OFILE)))
    ; Only if there's a file to load and the file exists.
  (COND
    ((AND GETFN (EQ (APPLY* GETFN (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
      (FULLNAME OFILE)
      'BEFORE)
      'DON'T))
      ; He doesn't want this document put. Bail out.
    (RETURN)))
  (TEXTPROP TEXTOBJ 'LASTGETFILENAME NIL)
  (RESETLST
    (RESETSAVE (TTYDISPLAYSTREAM (OR (AND (NEQ (fetch (TEXTOBJ PROMPTWINDOW) of TEXTOBJ)
      'DON'T)
      (fetch (TEXTOBJ PROMPTWINDOW) of TEXTOBJ))
      PROMPTWINDOW)))
    (RESETSAVE (CURSOR WAITINGCURSOR))
    (\SHOWSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
      NIL NIL)
    (\TEXTCLOSEF (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ))
    ; CLOSE the old files
  [OR (AND (STREAMP FILE)
    (OPENP FILE))
    (SETQ OFILE (OPENSTREAM OFILE 'INPUT) ; And open the new one.
  (SETQ PCTB (replace (TEXTOBJ PCTB) of TEXTOBJ with (TEDIT.BUILD.PCTB OFILE TEXTOBJ NIL NIL
    (fetch (TEXTOBJ DEFAULTCHARLOOKS)
    of TEXTOBJ)
    (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)
    UNFORMATTED?)))
  (for FORM in TEDIT.GET.FINISHEDFORMS do (EVAL FORM))
  ; Do any necessary cleanup for outside packages
  (SETQ LINES (fetch (TEXTOBJ LINES) of TEXTOBJ))
  (replace (TEXTOBJ \DIRTY) of TEXTOBJ with NIL)
  (for FIRSTLINE inside LINES do (replace (LINEDESCRIPTOR NEXTLINE) of FIRSTLINE with NIL))
  (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL)
  ;; The old cached piece is no longer valid--keep people from stepping on it, to prevent lost type-in and smashing other
  ;; documnts to which it has been moved...
  ;; (replace TEXTLEN of TEXTOBJ with (SUB1 (\EDITLT PCTB (SUB1 (\EDITLT PCTB \PCTBLastPieceOffset))))
  (replace (TEXTOBJ TEXTLEN) of TEXTOBJ with (fetch (BTREENODE TOTLEN) of PCTB))
  (replace (SELECTION CH#) of SEL with (replace (SELECTION CHLIM) of SEL with 1))
  (replace (SELECTION DCH) of SEL with 0)
  (replace (SELECTION POINT) of SEL with 'LEFT)
  (replace (SELECTION SET) of SEL with T)
  (replace (SELECTION SET) of (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ) with NIL)
  (replace (SELECTION SET) of (fetch (TEXTOBJ SHIFTEDSEL) of TEXTOBJ) with NIL)
  (replace (SELECTION SET) of (fetch (TEXTOBJ MOVESEL) of TEXTOBJ) with NIL)
  (replace (SELECTION SET) of TEDIT.SELECTION with NIL)
  (replace (SELECTION SET) of TEDIT.SHIFTEDSELECTION with NIL)
  (replace (TEXTOBJ CARETLOOKS) of TEXTOBJ with (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL))
  (for WINDOW inside (fetch (TEXTOBJ \WINDOW) of TEXTOBJ) as LINE inside LINES
    do
      ; Fill the edit window (s) with the new text
      (\FILLWINDOW (fetch (LINEDESCRIPTOR YBOT) of LINE)
        LINE TEXTOBJ NIL WINDOW))
  (\FIXSEL SEL TEXTOBJ)
  (\SHOWSEL SEL NIL T)
  (SETQ TITLE (TEXTSTREAM.TITLE TEXTOBJ)) ; find and set the title
  (\TEDIT.WINDOW.TITLE TEXTOBJ (\TEDIT.ORIGINAL.WINDOW.TITLE TITLE NIL))
  (SETQ MENUSTREAM (TEDITMENU.STREAM TEXTOBJ))
  (COND
    ((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)))
    (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ (\TEDIT.PRIMARYW TEXTOBJ))
    (\TEDIT.HISTORYADD TEXTOBJ (create TEDITHISTORYEVENT
      THACTION _ 'Get)))
  (AND GETFN (APPLY* GETFN (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
    (FULLNAME (fetch (TEXTOBJ TXTFILE) of TEXTOBJ))

```

```

      'AFTER]
(OFILE (TEDIT.PROMPTPRINT TEXTOBJ "[File not found.]")
  (TEXTPROP TEXTOBJ 'LASTGETFILENAME OFILE) ; Remember the file name he tried for, so we offer it next time.
)
(T (TEDIT.PROMPTPRINT TEXTOBJ "[Get aborted.]" T))

```

(TEDIT.PARSE.PAGEFRAMES1

[LAMBDA (PAGELIST PARENT)

; Edited 2-Jan-87 12:21 by jds

(* Take an external pageframe and internalize it.)

```

(PROG (FRAMETYPE PAGEFRAME)
  (COND
    ((type? PAGERECTION PAGELIST)
      (RETURN PAGELIST))
    ((NEQ 'LIST (SETQ FRAMETYPE (pop PAGELIST)))
      [SETQ PAGEFRAME (create PAGERECTION
        REGIONFILLMETHOD _ FRAMETYPE
        REGIONTYPE _ (pop PAGELIST)
        REGIONLOCALINFO _ (pop PAGELIST)
        REGIONSPEC _ (for VAL in (OR (pop PAGELIST)
          (LIST 0 0 0 0))
        collect (\TEDIT.SCALE VAL (CONSTANT (FQUOTIENT 1 35.27778)
          TEDIT.PARSE.PAGEFRAMES1
          ALIST PAGEFRAME)))
      (RETURN PAGEFRAME))
    (T (RETURN (for FRAMESPEC in (CAR PAGELIST) collect (TEDIT.PARSE.PAGEFRAMES1 FRAMESPEC NIL]))

```

(\ARBIN

[LAMBDA (STREAM)

(* jds "13-Nov-86 20:21")

; Read an arbitrary object from a file, parse it, and return it.

```

(PROG ((LEN (\SMALLPIN STREAM))
  USERSTR)
  (COND
    ((NOT (ZEROP LEN))
      (SETQ USERSTR (OPENSTRINGSTREAM (\STRINGIN STREAM LEN)
        'INPUT))
      (RETURN (PROG1 (READ USERSTR *TEDIT-FILE-READTABLE*)
        (CLOSEF? USERSTR))

```

(\ATMIN

[LAMBDA (STREAM)

(* jds "3-Apr-84 10:41")

```

(PROG ((LEN (\SMALLPIN STREAM)))
  (RETURN (COND
    ((ZEROP LEN)
      NIL)
    (T (PACK (for I from 1 to LEN collect (CHARACTER (\BIN STREAM))

```

(\DWIN

[LAMBDA (FILE)

(* jds "3-JAN-83 16:08")

```

(IPLUS (LLSH (\BIN FILE)
  24)
  (LLSH (\BIN FILE)
  16)
  (LLSH (\BIN FILE)
  8)
  (\BIN FILE])

```

(\STRINGIN

[LAMBDA (STREAM SETLEN)

; Edited 20-Apr-88 19:54 by jds

(* Read a string in length-contents form%: One word for the length, and one byte per character contained. However, the length may be specified by the caller instead of being read from the file.)

```

(PROG ((LEN (OR SETLEN (\SMALLPIN STREAM)))
  STR)
  (SETQ STR (ALLOCSTRING LEN))
  [OR (ZEROP LEN)
    (for I from 1 to LEN do (RPLCHARCODE STR I (READCCODE STREAM))
  (RETURN STR)]

```

(TEDIT.FORMATTEDP1

[LAMBDA (FILE LEN)

; Edited 12-Feb-88 11:43 by jds

; Checks for a version-1 formatted file

;; Returns NIL if it isn't a formatted file, or the # of pieces needed if it is; leaves file at start of text or of piece descriptions, resp.

```

(SETQ LEN (OR LEN (GETEOFPPTR FILE)))
(PROG (DESCPTR NPIECES PASSWORD)
  (COND
    ((ILEQ LEN 8)
      (RETURN NIL))

```

; Too short to be formatted.


```

(T (SETFILEPTR FILE (IDIFFERENCE LEN 8)) ; Move to start of FILEPTR to descriptions
  (SETQ DESCPTR (\DWIN FILE)) ; Read the file pos of the descriptions
  (SETQ NPIECES (\SMALLPIN FILE))
  (SETQ PASSWORD (\SMALLPIN FILE))
  (COND
    ((IEQP PASSWORD 31418) ; Version 3 TEdit format; instituted on 5/22/85
      (SETFILEPTR FILE DESCPTR)
      (RETURN NPIECES))
    ((IEQP PASSWORD 31417)
      ;; Version 2 format. Obsoleted 5/22/85 to permit revision of looks in the future without loss of compatibility
      (SETFILEPTR FILE DESCPTR)
      (RETURN (CONS 2 NPIECES)))
    ((IEQP PASSWORD 31416) ; VERSION 1 TEDIT FORMAT
      (SETFILEPTR FILE DESCPTR)
      (RETURN (CONS 1 NPIECES)))
    ((IEQP PASSWORD 31415) ; VERSION 0 TEDIT FORMAT
      (SETFILEPTR FILE DESCPTR)
      (RETURN (CONS 0 NPIECES)))
    (T ; NOT A FORMATTED FILE
      (SETFILEPTR FILE 0)
      (RETURN NIL))
  )

```

(\TEDIT.SET.WINDOW

[LAMBDA (TOWIND)

; Edited 12-Jun-90 18:16 by mitani
 (* USED IN RESETSAVES TO NULL OUT A TEXTSTREAM'S
 WINDOW BRIEFLY.)

```

(PROG1 (CONS (CAR TOWIND)
  (fetch (TEXTOBJ \WINDOW) of (CAR TOWIND)))
  (replace (TEXTOBJ \WINDOW) of (CAR TOWIND) with (CDR TOWIND))))

```

)

;; INCLUDEing a file

(DEFINEQ

(\TEDIT.INCLUDE

[LAMBDA (STREAM FILE START END SAFE)

; Edited 4-May-95 10:29 by sybalsky:mv:envos

;; Obtain a file name, and include that file's contents at the place where the caret is.

;; Returns T if the insertion happened, NIL if there was no place to put it.

```

(SETQ STREAM (TEXTOBJ STREAM))
(PROG ((SEL (fetch (TEXTOBJ SEL) of STREAM))
  PCTB TEXTLEN NFILE NFILE INSERTCH# INSPC LEN INSPC# PCLST NPC WASOPEN PCCOUNT NSTREAM START-OF-PIECE
  )
  (DECLARE (SPECVARS START-OF-PIECE))
  (COND
    ((fetch (TEXTOBJ TXTREADONLY) of STREAM) ; This is read-only.
      )
    ((fetch (SELECTION SET) of SEL) ; There is a place to do the include.
      [SETQ NFILE (OR FILE (\TEDIT.MAKEFILENAME (TEDIT.GETINPUT STREAM "Name of the file to load: ")
      (COND
        ((NOT NFILE) ; If no file was given, don't bother INCLUDEing.
          (TEDIT.PROMPTPRINT STREAM "[Include aborted.]" T)
          (RETURN))
        ((STREAMP NFILE))
        ((NOT (INFILEP NFILE)) ; Can't find the file. Put out a message.
          (TEDIT.PROMPTPRINT STREAM "[File not found.]" T)
          (RETURN))
        (COND
          ((NOT SAFE)
            ;; If the caller sets SAFE, we don't need to do any of this copying, because he's guaranteeing that the files'll be there until we
            ;; don't need 'em any more.
            (SETQ NFILE (OPENSTREAM '{NODIRCORE} 'OUTPUT 'NEW)) ; Create the holding file
            [SETQ NFILE (COND
              ((OPENP NFILE)
                (SETQ WASOPEN T)
                NFILE)
              (T
                (OPENFILE NFILE 'INPUT) ; Wasn't open -- need to open it for input...
                ; And copy the file-section into it.
                [COPYCHARS NFILE NFILE (OR START 0)
                  (OR END (GETFILEINFO NFILE 'LENGTH)
                ; Have to explicitly fill in 0 and EOFPTR, because if the file was
                ; open already, NILs would only copy from current fileptr to EOF.
                ; If the file didn't come to use open, close it.
                (OR WASOPEN (CLOSEF NFILE))
                (CLOSEF NFILE)
                (SETQ NFILE NFILE)
                (SETQ START (SETQ END NIL)) ; Then pretend nothing happened.
                ))
              (TEDIT.DO.BLUEPENDINGDELETE SEL STREAM) ; Delete any text, if need be
              (SETQ TEXTLEN (fetch (TEXTOBJ TEXTLEN) of STREAM)) ; We need the POST-deletion text length for later, so this must
              ; come after the b-p-d.
            ]
          )
        )
      )
    )

```

```

(\SHOWSEL SEL NIL NIL) ; Turn off SELs before we go any further
[SETQ NFILE (TEXTOBJ (SETQ NSTREAM (OPENTEXTSTREAM (OPENSTREAM NFILE 'INPUT)
NIL NIL NIL (LIST 'FONT (\TEDIT.GET.INSERT.CHARLOOKS
STREAM SEL)
'PARALOOKS
(fetch (TEXTOBJ FMTSPEC) of STREAM]

;; Get a textobj to describe the include source file (need NSTREAM so that if we have to convert it to formatted, we won't have lost
;; the textstream--and thus smash the free list.)

(COND
  ((AND (fetch (TEXTOBJ FORMATTEDP) of NFILE)
        (NOT (fetch (TEXTOBJ FORMATTEDP) of STREAM)))
    ; If the included text is formatted but this file isn't, let's format it!
    (\TEDIT.CONVERT.TO.FORMATTED STREAM))
  ((AND (fetch (TEXTOBJ FORMATTEDP) of STREAM)
        (NOT (fetch (TEXTOBJ FORMATTEDP) of NFILE)))
    ; The TARGET document is formatted, but the INCLUDED text isn't. Better format it before completing the include.
    (\TEDIT.CONVERT.TO.FORMATTED NFILE)))
  (SETQ PCTB (fetch (TEXTOBJ PCTB) of STREAM)) ; HERE, because the conversion to formatted will lengthen the
; pctx

[SETQ INSERTCH# (COND
  ((EQ (fetch (SELECTION POINT) of SEL)
        'LEFT)
    (fetch (SELECTION CH#) of SEL))
  (T (fetch (SELECTION CHLIM) of SEL))
) ; Find the place to make the insertion.
; The piece to make the insertion in

(SETQ INSPC (\CHTOPC INSERTCH# PCTB T))
[COND
  ((NEQ INSPC 'LASTPIECE)
    (COND
      ((IGREATERP INSERTCH# START-OF-PIECE) ; Must split the piece.
        (SETQ INSPC (\SPLITPIECE INSPC INSERTCH# STREAM INSPC#))
        (SETQ PCTB (fetch (TEXTOBJ PCTB) of STREAM)) ; Refresh the PCTB in case it grew.
      )
    )
  )
(SETQ PCLST (fetch (TEXTOBJ PCTB) of NFILE)) ; A temporary pctx, holding the pieces which describe the
; INCLUDED text

(SETQ LEN (fetch (BTREENODE TOTLEN) of PCLST))
(\TEDIT.INSERT.PIECES STREAM INSERTCH# (SETQ PCLST (\GETBASEPTR (\FIRSTNODE PCLST)
0))
LEN INSPC INSPC# NIL)
[COND
  ((AND (fetch (TEXTOBJ FORMATTEDP) of STREAM)
        (NOT (fetch (TEXTOBJ FORMATTEDP) of NFILE)))
    ; If the included text is formatted but this file isn't, let's format it!
    (\TEDIT.CONVERT.TO.FORMATTED STREAM INSERTCH# (IPLUS INSERTCH# LEN)
    (\TEDIT.HISTORYADD STREAM (create TEDITHISTOREVENT
THACTION _ 'Include
THCH# _ INSERTCH#
THLEN _ LEN
THFIRSTPIECE _ PCLST))
    ; Remember that we did this, so it can be undone.
(replace (TEXTOBJ TEXTLEN) of STREAM with (IPLUS TEXTLEN LEN))
    ; Inserting the pieces didn't fix up things like the length of the
; document, so do it now.

(AND (fetch (TEXTOBJ \WINDOW) of STREAM)
  (\FIXILINES STREAM SEL INSERTCH# LEN TEXTLEN))
  ; Mark any changed lines dirty.
(replace (SELECTION CHLIM) of SEL with (IPLUS (replace (SELECTION CH#) of SEL with INSERTCH#
LEN)) ; Now fix up the selection to be the included text, point_left,
; character selection grain.

(replace (SELECTION DCH) of SEL with LEN)
(replace (SELECTION DX) of SEL with 0)
(replace (SELECTION POINT) of SEL with 'RIGHT) ; So that several things INCLUDED in sequence fall in sequence.
(replace (SELECTION SELKIND) of SEL with 'CHAR)
(replace (SELECTION SELOBJ) of SEL with NIL)
(COND
  ((fetch (TEXTOBJ \WINDOW) of STREAM)
    ; We're displaying; update the display and the selection's line
; references
    (TEDIT.UPDATE.SCREEN STREAM)
    (\FIXSEL SEL STREAM)
    (\SHOWSEL SEL NIL T)))
  (replace (TEXTOBJ \DIRTY) of STREAM with T) ; Mark the document changed
  (\SETUPGETCH (IPLUS -1 INSERTCH# LEN (- (fetch (TEXTOBJ TEXTLEN) of STREAM)
(+ TEXTLEN LEN)))
    STREAM) ; Set the fileptr to the end of the insertion; the (- fetch ...) form
; accounts for any change due to NSCHAR translation that may
; occur during the screen update.

T)
(T (TEDIT.PROMPTPRINT STREAM "Please choose the place for the INCLUDE first." T))

```

(TEDIT.RAW.INCLUDE

```

[LAMBDA (STREAM INFILE START END) ; Edited 27-May-93 16:36 by sybalsky:mv:envos

```

```

;; takes a text stream and an OPEN stream to include. Note: Start and End are inclusive ptrs, unlike in copybytes and friends

```

[illegible]

```
(SETQ FONTFILEUSED (COND
```

```

(OLDFORMAT? (TEDIT.PUT.PCTB2 TEXTOBJ OFILE UNFORMATTED?))
(T (TEDIT.PUT.PCTB TEXTOBJ OFILE UNFORMATTED?])
(CLOSEF OFILE) ; Close the file, to free it up. And re-open it for INPUT only
[COND
  ((NOT CACHE) ; CSI! if caching do not need to reopen the output file anyway
   (SETQ OFILE (OPENSTREAM (fetch (STREAM FULLFILENAME) of OFILE)
    'INPUT] ; changed TEMPORary for ns filing with caching. may not work
            ; in general
(CLOSEF? (fetch (TEXTOBJ TXTFILE) of TEXTOBJ)) ; Close the old text file
(replace (TEXTOBJ TXTFILE) of TEXTOBJ with OFILE) ; And remember the new one for next time.
(replace (TEXTOBJ \DIRTY) of TEXTOBJ with NIL) ; We can safely QUIT now without losing anything.
)
(SETQ CH#S (REVERSE (CDR FONTFILEUSED))) ; The true filepos's of the pieces in the output file.
[COND
  ((AND (NOT CACHE)
        (RANDACCESSP OFILE)
        (EQ CR.EOLC (fetch (STREAM EOLCONVENTION) of OFILE)))
   ;; If we've cached this file, DON'T go thru and fill in the real file's location, because the EOL convention may well be wrong.
   ;; (SETQ PC (ELT (fetch PCTB of TEXTOBJ) (ADD1 \FirstPieceOffset)))
   (UNINTERRUPTABLY
    (SETQ PC (\GETBASEPTR (\FIRSTNODE (fetch (TEXTOBJ PCTB) of TEXTOBJ)
      0))
    (while (AND PC CH#S) do ;; Run thru the pieces in the PCTB, pointing them to the new file and their new locations.
      (COND
        ((fetch (PIECE POBJ) of PC))
        (T (replace (PIECE PFPOS) of PC with (pop CH#S))
         (CLOSEF? (fetch (PIECE PFILE) of PC))
         ; If this is a piece on an open file, close it, since we're never
         ; going to read from it again.
         (replace (PIECE PFILE) of PC with OFILE)
         (replace (PIECE PSTR) of PC with NIL)))
        (SETQ PC (fetch (PIECE NEXTPIECE) of PC))))])
(TEDIT.PROMPTPRINT TEXTOBJ "done.") ; Tell him we're finished.
(SETQ TITLE (TEXTSTREAM.TITLE TEXTOBJ)) ; find and set the title
(\TEDIT.WINDOW.TITLE TEXTOBJ (\TEDIT.ORIGINAL.WINDOW.TITLE TITLE NIL))
(SETQ MENUSTREAM (TEDITMENU.STREAM TEXTOBJ))
(COND
  ((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))
  (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL) ; Make sure any new insertions happen for real, and not as
   ; appends. Since all the pieces now point to the file rather than
   ; the strings.
  (replace (TEXTOBJ \INSERTPC) of TEXTOBJ with NIL)
  ;; make sure that TEDIT doesn't try to just add to the \INSERTPC since it will now have a pfile property
  (\TEDIT.HISTORYADD TEXTOBJ (create TEDIT.HISTORYEVENT
    THACTION _ 'Put
    THCH# _ 0
    THLEN _ 0
    THFIRSTPIECE _ NIL)) ; Remember we did this.
  (AND PUTFN (APPLY* PUTFN (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
    (fetch (STREAM FULLNAME) of (fetch (TEXTOBJ TXTFILE) of TEXTOBJ))
    'AFTER)) ; CSI! changed to not presume ofile is the txtfile anymore
  (RETURN OFILE]))

```

(TEDIT.PUT.PCTB

```

[LAMBDA (TEXTOBJ OFILE UNFORMATTED? SEPARATEFORMAT) ; Edited 27-May-93 16:00 by sybalsky:mv:envos
  ;; Put a representation of the piece table onto OFILE, preserving font changes and paragraph looks. UNFORMATTED? means write no font or
  ;; formatting info.
  (PROG (OCURS CH PC PFILE PSTR POBJ OFILELEN OLDLOOKS (OLDPARALOOKS (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
    (*READTABLE* *TEDIT-FILE-READTABLE*)
    (*PRINT-BASE* 10)
    OLDCH# CURCH# PREVPC FONTFILE (PCCOUNT 0)
    TRUEFILE CHARLOOKSLST PARALOOKSLST (TEDIT.PUT.FINISHEDFORMS NIL)
    (EDITSTENTATIVE (TEXTPROP TEXTOBJ 'TEDIT.TENTATIVE))
    (PARALOOKSSEEN NIL)
    (FORMATTINGLEVEL (TEDIT.FORMATTEDFILEP TEXTOBJ))
    (CACHE (TEXTPROP TEXTOBJ 'CACHE))
    CH#S PREVFPAT PARAHASH LOOKSHASH PREVPREVPC)
    (replace (STREAM LINELENGTH) of OFILE with MAX.SMALLP) ; Prevent spurious carriage-returns in the piece descriptions.
  ;; (SETQ PC (\EDITELT (fetch PCTB of TEXTOBJ) (ADD1 \FirstPieceOffset)))
  (SETQ PC (\GETBASEPTR (\FIRSTNODE (fetch (TEXTOBJ PCTB) of TEXTOBJ)
    0)) ; First piece in the document
  (SETQ OLDLOOKS (OR (AND (type? PIECE PC)
    (fetch (PIECE PLOOKS) of PC))
    (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
    TEDIT.DEFAULT.CHARLOOKS)) ; Starting looks
  (COND

```

```

((NEQ (fetch (STREAM EOLCONVENTION) of OFILE)
  CR.EOLC) ; This file is on a non-CR host; make a note to cache it
  (SETQ TRUEFILE OFILE) ; Remember where the file should wind up.
  (SETQ OFILE (OPENSTREAM '{NODIRCORE}' BOTH 'NEW)) ; And open a temp file to write it to.
  (replace (STREAM LINELENGTH) of OFILE with MAX.SMALLP) ; Prevent spurious carriage-returns in the piece descriptions.
))
[SETQ CURCH# (SETQ OLDCH# (ADD1 (GETFILEPTR OFILE)
(COND
  ((fetch (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ) ; There is layout info for this file. Save it
  (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE)) ; Open a font-info file if one is needed.
  (TEDIT.PUT.PAGEFRAMES FONTFILE (fetch (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ))
  (add PCCOUNT 1)))
  (\TEDIT.FLUSH.UNUSED.LOOKS TEXTOBJ PC) ; Run thru the lists of char & para looks and remove any that
  ; aren't in use
(COND
  ([AND (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ)
    (OR (IGREATERP (FLENGTH (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
      1)
      (NOT (EQFMTSPEC (CAR (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
        TEDIT.DEFAULT.FMTSPEC)
        ;; There are paragraph looks in this document that don't match the default -- save the list of them for later retrieval.
        (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE)) ; Open a font-info file if one is needed.
        (SETQ PARAHASH (\TEDIT.PUT.PARALOOKS.LIST FONTFILE (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
        (SETQ PARALOOKSSEEN T)))
    (COND
      ((OR PARALOOKSSEEN FORMATTINGLEVEL)
        ;; There are character looks in this document that don't match the default (or paragraph formatting, which forces looks to be saved) --
        ;; save the list for later retrieval.
        (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE)) ; Open a font-info file if one is needed.
        (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ)
[while PC do (COND
  ([AND (NOT (ZEROP (fetch (PIECE PLEN) of PC)))
    (OR (NOT PREVPC)
      (fetch (PIECE PPARALAST) of PREVPC))
    (OR PARALOOKSSEEN (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
      (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ]
      ; The last piece ended a paragraph, so send out new para looks
      (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE))
      ; Open a font-info file if one is needed.
      (COND
        ((NEQ CURCH# OLDCH#) ; There were prior characters that hadn't been described in a
        ; piece yet. Describe them
        [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE
          (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ]
          (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDOLOOKS PREVPC EDITSTENTATIVE
            LOOKSHASH PREVFPATP)
          (add PCCOUNT 1)
          (SETQ OLDCH# CURCH#) ; And now we've described all the characters up to the current
          ; one.
          ))
          (\TEDIT.PUT.PARALOOKS FONTFILE PC PARAHASH)
          (SETQ PARALOOKSSEEN T) ; Remember that we've seen a foreign paraloooks, and must
          ; henceforth note para boundaries
          (add PCCOUNT 1)))
      (COND
        [ (fetch (PIECE POBJ) of PC) ; It's an object -- go use its PUTFN
        (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE)) ; Open a font-info file if one is needed.
        (COND
          ((AND (NEQ CURCH# OLDCH#)
            PREVPC) ; There were prior characters that hadn't been described in a
            ; piece yet. Describe them
            [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE
              (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ]
              (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDOLOOKS PREVPC EDITSTENTATIVE
                LOOKSHASH PREVFPATP)
              (add PCCOUNT 1)
              (SETQ OLDCH# CURCH#) ; And now we've described all the characters up to the current
              ; one.
              ))
              ; If the prior thing was text, send along its descriptor.
              (AND (NOT UNFORMATTED?)
                (add CURCH# (TEDIT.PUT.OBJECT PC OFILE FONTFILE CURCH#)))
                ; Send out the object
                (add PCCOUNT 1)
                (SETQ OLDCH# CURCH#)
                (COND
                  ([OR (NOT (EQCLOOKS OLDOLOOKS (fetch (PIECE PLOOKS) of PC)))
                    (NEQ (fetch (PIECE PFATP) of PC)
                      (AND PREVPC (fetch (PIECE PFATP) of PREVPC)))
                    [AND EDITSTENTATIVE (NEQ (fetch (PIECE PNEW) of PC)
                      (AND PREVPC (fetch (PIECE PNEW) of PREVPC)

```

```

        (AND (OR (NOT PREVPC)
                  (fetch (PIECE PPARALAST) of PREVPC))
              (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
                            (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)
                            ; The OBJECT has different looks from before
                            )
              )
        (\BOUT FONTFILE 1)
        (\TEDIT.PUT.SINGLE.CHARLOOKS FONTFILE (fetch (PIECE PLOOKS) of PC))
        (SETQ OLDLOOKS (fetch (PIECE PLOOKS) of PC))
        (T
          (\BOUT FONTFILE 0)
          ; No differences. Don't write any charlooks, and mark that fact
          ; MAKE BLOODY SURE THAT THE NEXT RUN OF
          ; CHARACTERS GETS ITS OWN LOOKS
        )
      ]
    ((NOT (ZEROP (fetch (PIECE PLEN) of PC)))
     ; It's not an object.

    ;; For 0-length pieces, don't even acknowledge their existence!!
    ;; So only do this processing if there's text in the piece.
    [COND
      ([OR [NEQ (fetch (PIECE PFATP) of PC)
                (SETQ PREVFPATP (AND PREVPC (fetch (PIECE PFATP) of PREVPC)
              (NOT (EQCLOOKS OLDLOOKS (fetch (PIECE PLOOKS) of PC)))
              [AND EDITSTENTATIVE (NEQ (fetch (PIECE PNEW) of PC)
                (AND PREVPC (fetch (PIECE PNEW) of PREVPC)
              (AND (OR (NOT PREVPC)
                        (fetch (PIECE PPARALAST) of PREVPC))
                (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
                              (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)
                              ; We have a piece with new looks.
                              ; The PREVFPATP clause needs to come first, so that
                              ; PREVFPATP gets set for later use.
                (SETQ FONTFILE (\TEDIT-OPEN-FONT-FILE FONTFILE))
                ; Open a font-info file if one is needed.
              (COND
                ((NOT (IEQP OLDCH# CURCH#))
                 ; If there were looks past, and if the run was not empty, save a
                 ; piece for its looks
                 [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE
                                      (fetch (TEXTOBJ TXTCHARLOOKSLIST)
                                      of TEXTOBJ]
                  (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDLOOKS PREVPC EDITSTENTATIVE
                    LOOKSHASH PREVFPATP)
                    (add PCCOUNT 1)))
                  (SETQ OLDLOOKS (fetch (PIECE PLOOKS) of PC))
                  (SETQ OLDCH# CURCH#)
                  (COND
                    [PREVFPATP (COND
                      ((fetch (PIECE PFATP) of PC))
                      (T
                        ; Switching from FAT to thin
                        (BOUT OFILE 255)
                        (BOUT OFILE 0)
                        (add CURCH# 2]
                      ((fetch (PIECE PFATP) of PC)
                       ; Switching from thin to fat
                       (BOUT OFILE 255)
                       (BOUT OFILE 255)
                       (BOUT OFILE 0)
                       (add CURCH# 3]
                    ; Now dump out the non-object contents of the piece.
                  (COND
                    [(SETQ PFILE (fetch (PIECE PFILE) of PC))
                     ; It's on a file. Copy it.
                    (OR (OPENP PFILE)
                        (replace (PIECE PFILE) of PC with (SETQ PFILE (\TEDIT.REOPEN.STREAM TEXTOBJ
                                      PFILE]
                        ; Make sure the file is open.
                    (COPYBYTES PFILE OFILE (fetch (PIECE PFPOS) of PC)
                    (IPLUS (fetch (PIECE PFPOS) of PC)
                    (COND
                      ((fetch (PIECE PFATP) of PC)
                       ; For fat file pieces, copy twice as many bytes as characters.
                       (UNFOLD (fetch (PIECE PLEN) of PC)
                        2))
                      (T (fetch (PIECE PLEN) of PC]
                    ((SETQ PSTR (fetch (PIECE PSTR) of PC))
                     ; It's in a string. Just print it.
                    (COND
                      [(fetch (PIECE PFATP) of PC)
                       ; The string is fat: Copy twice as many bytes as chars.
                       (for I from 1 to (fetch (PIECE PLEN) of PC) as CH instring PSTR
                        do (\BOUT OFILE (\CHARSET CH))
                        do (\BOUT OFILE (\CHAR8CODE CH))
                      (T
                       ; The string is thin. Just copy it to the file.
                       (for I from 1 to (fetch (PIECE PLEN) of PC) as CH instring PSTR
                        do (\BOUT OFILE CH]
                    (COND
                      ((AND (NOT CACHE)
                           (RANDACCESSP OFILE))
                       ; CSLI leave the pieces and the pctb alone and just write the file
                       ; if its cached or not randomaccess
                      (push CH#S (SUB1 CURCH#]
                    (COND

```

```

    ( (fetch (PIECE PFATP) of PC)
      (add CURCH# (UNFOLD (fetch (PIECE PLEN) of PC)
                           2)))
    (T (add CURCH# (fetch (PIECE PLEN) of PC)
                  ; Keep running track of where in the file we are.
    ))
  (COND
    ((NOT (ZEROP (fetch (PIECE PLEN) of PC)))
     ;; Only remember this piece if it's not zero-length!
     (SETQ PREVPREVP PC)
     (SETQ PREVP PC))
    (SETQ PC (fetch (PIECE NEXTPIECE) of PC))
  )
  finally
    (COND
      ((AND FONTFILE (NEQ OLDCH# CURCH#))
       ; Only if there WERE characters, and only if there's a need for
       ; font information
       [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE (fetch (TEXTOBJ
                                                                                     TXTCHARLOOKSLIST
                                                                                     )
                                                                                     of TEXTOBJ)
                                PREVPREVP PC)
        ; Put out a description of the characters
        (add PCCOUNT 1))
      (COND
        ((AND PARALOOKSSEEN (fetch (PIECE PPARALAST) of PREVP))
         ; The last piece contained the end of a paragraph. Make sure it
         ; gets noted.
         (\TEDIT.PUT.PARALOOKS FONTFILE PREVP PARASHASH)
         ;; Write out a dummy paragraph-looks piece, so that we protect the PPARALAST of the final piece in the document.
         (\DWOUT FONTFILE 0)
         (\SMALLPOUT FONTFILE \PieceDescriptorPARA)
         (\SMALLPOUT FONTFILE 1)
         ;; This adds a total of 2 pieces to the file:
         (add PCCOUNT 2]
    (for FORM in TEDIT.PUT.FINISHEDFORMS do (EVAL FORM)) ; Do any user-specific cleanup
  (COND
    (TRUEFILE
     (COND
       ((AND FONTFILE (NOT UNFORMATTED?)
              (NOT SEPARATEFORMAT))
        ; Formatted file: Copy without converting.
        (COPYBYTES OFFILE TRUEFILE 0 -1))
       (T
        ; Go ahead and convert the EOLCONVENTION, this is a
        ; plain-text file
        (COPYCHARS OFFILE TRUEFILE 0 -1)))
      (SETQ OFFILE TRUEFILE)))
  [COND
    ((AND (OPENP OFFILE)
          FONTFILE)
     ; We need to write format info.
     ; So remember the end of the plain-text part of the file
     ; # OF PIECES WE" NEED TO RECONSTRUCT THIS FILE
     ; Now the password for NEW format files: 31416
     (\DWOUT FONTFILE (GETFILEPTR OFFILE))
     (\SMALLPOUT FONTFILE PCCOUNT)
     (\SMALLPOUT FONTFILE 31418)
     (COND
       ((AND (NOT UNFORMATTED?)
            (NOT SEPARATEFORMAT))
        ;; Only write fmtg info at the end if we want it there--not if we want plain text or want it kept separate.
        (COPYBYTES FONTFILE OFFILE 0 (GETEOFPTR FONTFILE))
        ; Copy the font information to the file trailer
        )
       (T)
     (CLOSEF FONTFILE)
     (COND
       ((NOT SEPARATEFORMAT)
        ; Unless we want the formatting info separately, delete the file
        ; (since FONTFILE is a stream, we should not need to delete it at
        ; all) (DELFILE FONTFILE)
        )
     ]
  (replace (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ with (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ
                                                                                     DEFAULTCHARLOOKS
                                                                                     )
                                                                                     of TEXTOBJ)
                                TEXTOBJ))
  ; Re-add the default and caret looks's to the lists, since they may
  ; not have been really saved.
  (replace (TEXTOBJ CARETLOOKS) of TEXTOBJ with (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ CARETLOOKS)
                                                                                     of TEXTOBJ)
                                TEXTOBJ))
  (replace (TEXTOBJ FMTSPEC) of TEXTOBJ with (\TEDIT.UNIQIFY.PARALOOKS (fetch (TEXTOBJ FMTSPEC)
                                                                                     of TEXTOBJ)
                                TEXTOBJ))
  (RETURN (CONS (COND
    (UNFORMATTED? NIL)

```



```
(T FONTFILE))
CH#S])
```

(\TEDIT.PUTRESET

```
[LAMBDA (PROC&VALUE)
  (CONS (CAR PROC&VALUE)
    (PROCESSPROP (CAR PROC&VALUE)
      'BEFOREEXIT
      (CDR PROC&VALUE]))
```

(* jds "15-May-85 16:38")

(TEDIT.PUT.PIECE.DEScriptor

```
[LAMBDA (FILE CH1 CHLIM LOOKS)
```

```
; Edited 30-May-91 20:25 by jds
(* Put a description of LOOKS into FILE.
LOOKS apply to characters CH1 thru CHLIM-1)
(* (PROG ((FONT (fetch (CHARLOOKS CLFONT) of LOOKS))
STR) (SETQ STR (CONCAT "(FONTCREATE "
(KWOTE (FONTPROP FONT (QUOTE FAMILY))) " "
(FONTPROP FONT (QUOTE SIZE))) " "
(KWOTE (FONTPROP FONT (QUOTE FACE))) " ")")
(\DWOUT FILE (IDIFFERENCE CHLIM CH1))
(* The length of this run of looks)
(\SMALLPOUT FILE (NCHARS STR))
(* The length of the description which follows)
(PRIN1 STR FILE) (* Print the form which can EVAL to re-create
TECTED) of LOOKS 8) (T 0)) (COND
((fetch (CHARLOOKS CLINVISIBLE) of LOOKS) NIL 4)
(T 0)) (COND ((fetch (CHARLOOKS CLSELHERE) of LOOKS)
2) (T 0)) (COND ((fetch (CHARLOOKS CLCANCOPY) of
LOOKS) 1) (T 0))))))
```

```
the font information) (\BOUT FILE (LOGOR (COND ((fetch (CHARLOOKS CLPRO
```

```
(HELP])
```

(\ARABOUT

```
[LAMBDA (STREAM ITEM)
```

```
; Edited 20-Apr-88 19:55 by jds
; Write an arbitrary MKSTRING-able thing in length-contents
; form.
```

```
(LET ((SIZE (AND ITEM (NCHARS ITEM T *TEDIT-FILE-READTABLE*)))
  (FPTR)
  (END-FPTR))
  (\SMALLPOUT STREAM (OR SIZE 0))
  (SETQ FPTR (GETFILEPTR STREAM))
  (OR (NOT ITEM)
    (ZEROP SIZE)
    (PRIN2 ITEM STREAM *TEDIT-FILE-READTABLE*)))
```

```
:: Because of NS chars, you gotta back up and really count bytes.
```

```
(* (SETQ END-FPTR (GETFILEPTR STREAM))
(SETFILEPTR STREAM FPTR) (\SMALLPOUT STREAM
(-
END-FPTR FPTR)) (SETFILEPTR STREAM END-FPTR))
```

```
NIL])
```

(\ATMOUT

```
[LAMBDA (STREAM ATOM)
```

```
(* jds "30-Jan-85 14:46")
(* Write an atom's characters in length-contents form.)
```

```
(\SMALLPOUT STREAM (COND
  (ATOM (NCHARS ATOM))
  (T 0)))
(OR (NOT ATOM)
  (ZEROP (NCHARS ATOM))
  (for CH inatom ATOM do (\BOUT STREAM CH]))
```

(\DWOUT

```
[LAMBDA (FILE NUMBER)
  (\BOUT FILE (LOGAND 255 (LRSH NUMBER 24)))
  (\BOUT FILE (LOGAND 255 (LRSH NUMBER 16)))
  (\BOUT FILE (LOGAND 255 (LRSH NUMBER 8)))
  (\BOUT FILE (LOGAND 255 NUMBER])
```

(* jds "3-JAN-83 15:30")

(\STRINGOUT

```
[LAMBDA (STREAM STRING LEN)
```

(* jds "1-May-84 11:58")

```
(* Write a string on a file in length-contents form; one word for the length, and one byte per character contained.)
```

```
(SETQ LEN (OR LEN (NCHARS STRING)))
(\SMALLPOUT STREAM LEN)
(OR (ZEROP LEN)
  (for CH instring STRING as I from 1 to LEN do (\BOUT STREAM CH]))
```

(\TEDIT-OPEN-FONT-FILE

```
[LAMBDA (EXISTING-FONTFILE-IF-ANY)
```

; Edited 23-Sep-87 12:31 by jds

```
:: Open a font-information file for TEDIT PUT operation, if one doesn't exist already. Also set its linelength to effective infinity, so that we don't get
```

;; spurious CRs in the middle of formatting info.

;; The calling form must be (SETQ FOO (\TEDIT-OPEN-FONT-FILE FOO)), to preserve information.

```
(COND
  ((NOT EXISTING-FONTFILE-IF-ANY) ; Create the font-info file if it doesn't exist yet
   (SETQ EXISTING-FONTFILE-IF-ANY (OPENSTREAM ' {NODIRCORE} 'BOTH 'NEW))
   (replace (STREAM LINELENGTH) of EXISTING-FONTFILE-IF-ANY with MAX.SMALLP) ; Prevent spurious carriage-returns in the piece descriptions.
  ))
```

EXISTING-FONTFILE-IF-ANY))

)

(DEFINEQ

(\TEDIT.GET.CHARLOOKS.LIST

```
[LAMBDA (FILE)
  (* jds "28-Jan-85 17:50")
  (* Read the list of CHARLOOKSs from the file.)
  (for I from 1 to (\SMALLPIN FILE) collect (\TEDIT.GET.SINGLE.CHARLOOKS FILE])
```

(\TEDIT.GET.SINGLE.CHARLOOKS

```
[LAMBDA (FILE)
  ; Edited 30-May-91 20:25 by jds
  (* Read a set of CHARLOOKS from FILE)
```

```
(PROG* ((LOOKS (create CHARLOOKS))
  (FILEPOS (GETFILEPTR FILE))
  (LOOKSLEN (\SMALLPIN FILE))
  FONT STR NAME NAMELEN SIZE SUPER SUB PROPS STYLESTR USERSTR)
  (SETQ NAME (\ARBIN FILE)) (* The font name)
  (SETQ SIZE (\SMALLPIN FILE)) (* Size of the type, in points)
  (SETQ SUPER (\SMALLPIN FILE)) (* Superscripting distance)
  (replace (CHARLOOKS CLSTYLE) of LOOKS with (OR (\ARBIN FILE)
  0))
  (replace (CHARLOOKS CLUSERINFO) of LOOKS with (\ARBIN FILE))
  (SETQ PROPS (\SMALLPIN FILE))
  (with CHARLOOKS LOOKS [SETQ CLLEADER (NOT (ZEROP (LOGAND 2048 PROPS)
  [SETQ CLINVERTED (NOT (ZEROP (LOGAND 1024 PROPS)
  [SETQ CLBOLD (NOT (ZEROP (LOGAND 512 PROPS)
  [SETQ CLITAL (NOT (ZEROP (LOGAND 256 PROPS)
  [SETQ CLULINE (NOT (ZEROP (LOGAND 128 PROPS)
  [SETQ CLOLINE (NOT (ZEROP (LOGAND 64 PROPS)
  [SETQ CLSTRIKE (NOT (ZEROP (LOGAND 32 PROPS)
  [SETQ CLSMALLCAP (NOT (ZEROP (LOGAND 16 PROPS)
  [SETQ CLPROTECTED (NOT (ZEROP (LOGAND 8 PROPS)
  [SETQ CLINVISIBLE (NOT (ZEROP (LOGAND 4 PROPS)
  [SETQ CLSELHERE (NOT (ZEROP (LOGAND 2 PROPS)
  [SETQ CLCANCOPY (NOT (ZEROP (LOGAND 1 PROPS)
  (SETQ CLSIZE SIZE)
  (SETQ CLOFFSET SUPER))
  [replace (CHARLOOKS CLFONT) of LOOKS with (COND
    ((LISTP NAME)
      (* This was a font class. Restore it.)
      (FONTCLASS (pop NAME)
        NAME))
    ((AND NAME (NOT (ZEROP SIZE)))
      (FONTCREATE NAME SIZE (COND
        ((AND (fetch (CHARLOOKS CLBOLD)
          of LOOKS)
          (fetch (CHARLOOKS CLITAL)
            of LOOKS))
        'BOLDITALIC)
        ((fetch (CHARLOOKS CLBOLD)
          of LOOKS)
        'BOLD)
        ((fetch (CHARLOOKS CLITAL)
          of LOOKS)
        'ITALIC]

  (SETFILEPTR FILE (IPLUS FILEPOS LOOKSLEN))
  (RETURN LOOKS])
```

(\TEDIT.PUT.CHARLOOKS.LIST

```
[LAMBDA (FILE LOOKSLIST)
  (* jds " 5-Mar-85 15:58")
  (* Write the list of CHARLOOKSs into the font file.)
```

(* Returns a hasharray that will map from a given CHARLOOKS to that CHARLOOKS' position in the list we wrote on the file. Those position numbers are then written in the individual looks descriptions, and are used to reconstruct the piece looks when the file is read back in.)

```
(PROG ((LOOKSHASH (HASHARRAY 50)))
  (\DWOUT FILE 0) (* No characters are described by this pseudo-piece entry.)
  (\SMALLPOUT FILE \PieceDescriptorCHARLOOKSLIST) (* Mark it as containing the list of CHARLOOKSs)
  (\SMALLPOUT FILE (FLENGTH LOOKSLIST)) (* How many CHARLOOKSs there are in the list)
  (for I from 1 as LOOKS in LOOKSLIST do (* Write each charlooks, in the order they appear in the list.)
    (\TEDIT.PUT.SINGLE.CHARLOOKS FILE LOOKS)
    (* Write out the description)
```

(PUTHASH LOOKS I LOOKSHASH)

(* And save it in the hash table so people can find its index.))

(RETURN LOOKSHASH])

(\TEDIT.PUT.SINGLE.CHARLOOKS

[LAMBDA (FILE LOOKS)

; Edited 30-May-91 20:26 by jds

(* Put out a single CHARLOOKS description.)

```
(PROG ((FILEPOS (GETFILEPTR FILE))
      (FONT (fetch (CHARLOOKS CLFONT) of LOOKS))
      STR LEN)
```

(\SMALLPOUT FILE 0)

(* Reserve space for the length of this looks)

[COND

((type? FONTCLASS FONT)

(* For font classes, we need to save a list of device-FD sets)

(\ABOUT FILE (FONTCLASSUNPARSE FONT)))

(T

(\ATMOUT FILE (FONTPROP FONT 'FAMILY]

(* For FONTDESCRIPTORS, do it the easy way)

(* The font family)

```
(\SMALLPOUT FILE (OR (FONTPROP FONT 'SIZE)
                    0))
```

(* Size of the type, in points)

```
(\SMALLPOUT FILE (OR (fetch (CHARLOOKS CLOFFSET) of LOOKS)
                    0))
```

(* Super/subscripting distance)

[COND

([AND (fetch (CHARLOOKS CLSTYLE) of LOOKS)

(NOT (ZEROP (fetch (CHARLOOKS CLSTYLE) of LOOKS]

(\ABOUT FILE (fetch (CHARLOOKS CLSTYLE) of LOOKS)))

(T (\SMALLPOUT FILE 0)))

[COND

((fetch (CHARLOOKS CLUSERINFO) of LOOKS)

(\ABOUT FILE (fetch (CHARLOOKS CLUSERINFO) of LOOKS)))

(T (\SMALLPOUT FILE 0)))

[\SMALLPOUT FILE (LOGOR (COND

((fetch (CHARLOOKS CLEADER) of LOOKS)

(* Dotted-leader; relevant only to TABs)

2048)

(T 0))

[COND

((fetch (CHARLOOKS CLINVERTED) of LOOKS)

(* Inverse-video)

1024)

(T 0))

[COND

((fetch (CHARLOOKS CLBOLD) of LOOKS)

512)

(T 0))

[COND

((fetch (CHARLOOKS CLITAL) of LOOKS)

256)

(T 0))

[COND

((fetch (CHARLOOKS CLULINE) of LOOKS)

128)

(T 0))

[COND

((fetch (CHARLOOKS CLOLINE) of LOOKS)

64)

(T 0))

[COND

((fetch (CHARLOOKS CLSTRIKE) of LOOKS)

32)

(T 0))

[COND

((fetch (CHARLOOKS CLSMALLCAP) of LOOKS)

16)

(T 0))

[COND

((fetch (CHARLOOKS CLPROTECTED) of LOOKS)

8)

(T 0))

[COND

((fetch (CHARLOOKS CLINVISIBLE) of LOOKS)

NIL 4)

(T 0))

[COND

((fetch (CHARLOOKS CLSELHERE) of LOOKS)

2)

(T 0))

[COND

((fetch (CHARLOOKS CLCANCOPY) of LOOKS)

1)

(T 0]

(* * Now go fill in the length field at the front of the LOOKS. (ALL looks info should be written out BEFORE this comment.))

```
(SETQ LEN (IDIFFERENCE (GETFILEPTR FILE)
                      FILEPOS))
```

(* The length of this set of looks)

(SETFILEPTR FILE FILEPOS)

(* Go write the length field)

(\SMALLPOUT FILE LEN)

```

    (SETFILEPTR FILE -1)                                (* And back to the end of the file)
  ])
)

(DEFINEQ
(\TEDIT.GET.PARALOOKS.LIST
 [LAMBDA (FILE TEXTOBJ)
    (* jds "13-Jun-85 11:14")
    (* Read the list of CHARLOOKSs from the file.)
    (for I from 1 to (\SMALLPIN FILE) collect (\TEDIT.GET.SINGLE.PARALOOKS FILE TEXTOBJ)])

(\TEDIT.GET.SINGLE.PARALOOKS
 [LAMBDA (FILE TEXTOBJ)
    ; Edited 2-Jul-93 21:31 by sybalskY:MV:ENVOS
    ; Read a paragraph format spec from the FILE, and return it for
    ; later use.

    (PROG ((LOOKS (create FMTSPEC))
            (FILEPOS (GETFILEPTR FILE))
            (LOOKSLEN (\SMALLPIN FILE))
            TABFLG DEFAULTTAB TABCOUNT TABS TABSPEC TABTYPE QUAD)
      (replace (FMTSPEC 1STLEFTMAR) of LOOKS with (\SMALLPIN FILE))
                                ; Left margin for the first line of the paragraph
      (replace (FMTSPEC LEFTMAR) of LOOKS with (\SMALLPIN FILE))
                                ; Left margin for the rest of the paragraph
      (replace (FMTSPEC RIGHTMAR) of LOOKS with (\SMALLPIN FILE))
                                ; Right margin for the paragraph
      (replace (FMTSPEC LEADBEFORE) of LOOKS with (\SMALLPIN FILE))
                                ; Leading before the paragraph
      (replace (FMTSPEC LEADAFTER) of LOOKS with (\SMALLPIN FILE))
                                ; Lead after the paragraph
      (replace (FMTSPEC LINELEAD) of LOOKS with (\SMALLPIN FILE))
                                ; inter-line leading
      (replace (FMTSPEC TABSPEC) of LOOKS with (SETQ TABSPEC (CONS NIL NIL)))
                                ; Will be tab specs
      (SETQ TABFLG (\BIN FILE))
      (replace (FMTSPEC QUAD) of LOOKS with (SELECTC (SETQ QUAD (\BIN FILE))
                                                    (1 'LEFT)
                                                    (2 'RIGHT)
                                                    (3 'CENTERED)
                                                    (4 'JUSTIFIED)
                                                    (SHOULDN'T)))

      (COND
        ((NOT (ZEROP (LOGAND TABFLG 1)))
         (SETQ DEFAULTTAB (\SMALLPIN FILE))
         (SETQ TABCOUNT (\BIN FILE))
         [SETQ TABS (for TAB# from 1 to TABCOUNT collect (create TAB
                                                                TABX _ (\SMALLPIN FILE)
                                                                TABKIND _ (SELECTQ (SETQ TABTYPE (\BIN FILE))
                                                                (0 'LEFT)
                                                                (1 'RIGHT)
                                                                (2 'CENTERED)
                                                                (3 'DECIMAL)
                                                                (4 'DOTTEDLEFT)
                                                                (5 'DOTTEDRIGHT)
                                                                (6 'DOTTEDCENTERED)
                                                                (7 'DOTTEDDECIMAL)
                                                                (SHOULDN'T])
         (OR (ZEROP DEFAULTTAB)
              (RPLACA TABSPEC DEFAULTTAB))
              (RPLACD TABSPEC TABS)))

        [COND
          ((NOT (ZEROP (LOGAND TABFLG 2)))
           (replace (FMTSPEC FMTSPECIALX) of LOOKS with (\SMALLPIN FILE))
                                           ; There are tabs to read
                                           ; Special X location on page for this paragraph
           (replace (FMTSPEC FMTSPECIALY) of LOOKS with (\SMALLPIN FILE))
           (replace (FMTSPEC FMTUSERINFO) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTPARATYPE) of LOOKS with (\ATMIN FILE))
           (replace (FMTSPEC FMTPARASUBTYPE) of LOOKS with (\ATMIN FILE))
           (replace (FMTSPEC FMTSTYLE) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTCHARSTYLES) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTNEWPAGEBEFORE) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTNEWPAGEAFTER) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTHEADINGKEEP) of LOOKS with (\ARBIN FILE))
           (replace (FMTSPEC FMTKEEP) of LOOKS with (\ARBIN FILE))

           [COND
             ((ILESSP (GETFILEPTR FILE)
                      (IPLUS FILEPOS LOOKSLEN))
              (replace (FMTSPEC FMTBASETOBASE) of LOOKS with (\ARBIN FILE]

             [COND
               ((ILESSP (GETFILEPTR FILE)
                        (IPLUS FILEPOS LOOKSLEN))
                (replace (FMTSPEC FMTREVISED) of LOOKS with (\ARBIN FILE]

               [COND
                 ((ILESSP (GETFILEPTR FILE)
                          (IPLUS FILEPOS LOOKSLEN))
                  (replace (FMTSPEC FMTCOLUMN) of LOOKS with (\ARBIN FILE]

```

```

(COND
  ((ILESSP (GETFILEPTR FILE)
    (IPLUS FILEPOS LOOKSLEN))
    (replace (FMTSPEC FMTCHARSTYLES) of LOOKS with (\ARBIN FILE]
[COND
  ((ILESSP (GETFILEPTR FILE)
    (IPLUS FILEPOS LOOKSLEN))
    ; There is more PARALOOKS info in this piece -- we probably
    ; lost data.
    (TEDIT.PROMPTPRINT TEXTOBJ "WARNING: Newer file version; you lost PARALOOKS info" T)
    (SETFILEPTR FILE (IPLUS FILEPOS LOOKSLEN]
  (RETURN LOOKS])

```

(\TEDIT.PUT.PARALOOKS.LIST

[LAMBDA (FILE LOOKSLIST)

; Edited 1-Sep-87 20:34 by jds
; Write the list of FMTSPECS into the font file.

```

(PROG ((LOOKSHASH (HASHARRAY 50)))
  (\DWOUT FILE 0)
  (\SMALLPOUT FILE \PieceDescriptorPARALOOKSLIST)
  (\SMALLPOUT FILE (FLENGTH LOOKSLIST))
  (for I from 1 as LOOKS in LOOKSLIST do (\TEDIT.PUT.SINGLE.PARALOOKS FILE LOOKS)
    ; Write out the description
    (PUTHASH LOOKS I LOOKSHASH)
    ; And save it in the hash table so people can find its index.
  )
  (RETURN LOOKSHASH])

```

(\TEDIT.PUT.SINGLE.PARALOOKS

[LAMBDA (FILE LOOKS)

; Edited 2-Jul-93 21:30 by sybalsky:MV:ENVOS

;; Put a description of LOOKS into FILE.

```

(PROG ((FILEPOS (GETFILEPTR FILE))
  DEFAULTTAB TABSPECS OUTPUTFORMAT LEN)
  (\SMALLPOUT FILE 0)
  (\SMALLPOUT FILE (fetch (FMTSPEC 1STLEFTMAR) of LOOKS)) ; Reserve space for the length of this looks
  (\SMALLPOUT FILE (fetch (FMTSPEC LEFTMAR) of LOOKS)) ; Left margin for the first line of the paragraph
  (\SMALLPOUT FILE (fetch (FMTSPEC RIGHTMAR) of LOOKS)) ; Left margin for the rest of the paragraph
  (\SMALLPOUT FILE (fetch (FMTSPEC LEADBEFORE) of LOOKS)) ; Right margin for the paragraph
  (\SMALLPOUT FILE (fetch (FMTSPEC LEADBEFORE) of LOOKS)) ; Leading before the paragraph
  (\SMALLPOUT FILE (fetch (FMTSPEC LEADAFTER) of LOOKS)) ; Lead after the paragraph
  (\SMALLPOUT FILE (fetch (FMTSPEC LINELEAD) of LOOKS)) ; inter-line leading
  (SETQ DEFAULTTAB (CAR (fetch (FMTSPEC TABSPEC) of LOOKS)))
  (SETQ TABSPECS (CDR (fetch (FMTSPEC TABSPEC) of LOOKS)))
  (COND
    ((AND (fetch (FMTSPEC TABSPEC) of LOOKS)
      (OR DEFAULTTAB TABSPECS))
      ; There are tab specs to save, or there is a default tab setting to
      ; save
      (\BOUT FILE 3))
    (T
      ; There are no tab looks. Just let him go.
      (\BOUT FILE 2)))
  (\BOUT FILE (SELECTQ (fetch (FMTSPEC QUAD) of LOOKS)
    (LEFT 1)
    (RIGHT 2)
    ((CENTER CENTERED)
      3)
    ((JUST JUSTIFIED)
      4)
    (SHOULDNT)))
[COND
  ((OR TABSPECS DEFAULTTAB)
    ; There are tab specs to save.
    (COND
      (DEFAULTTAB (\SMALLPOUT FILE DEFAULTTAB))
      (T (\SMALLPOUT FILE 0)))
    (COND
      ((IGREATERP (LENGTH TABSPECS)
        255)
        (SHOULDNT "Paragraph has more than 255 TABs set--can't be saved."))
      (\BOUT FILE (LENGTH TABSPECS))
      (COND
        (TABSPECS
          ; # of tab settings <256!
          (for TAB in TABSPECS do (\SMALLPOUT FILE (fetch TABX of TAB))
            ; And setting.
            (\BOUT FILE (SELECTQ (fetch TABKIND of TAB)
              (LEFT 0)
              (RIGHT 1)
              (CENTERED 2)
              (DECIMAL 3)
              (DOTTEDLEFT 4)
              (DOTTEDRIGHT 5)
              (DOTTEDCENTERED
                6)
              (DOTTEDDECIMAL
                7)
              (SHOULDNT)))
            ; Tab type

```

```

(\SMALLPOUT FILE (OR (fetch (FMTSPEC FMTSPECIALX) of LOOKS)
0))
(\SMALLPOUT FILE (OR (fetch (FMTSPEC FMTSPECIALY) of LOOKS)
0))
(\ARBOUT FILE (fetch (FMTSPEC FMTUSERINFO) of LOOKS))
(\ATMOUT FILE (fetch (FMTSPEC FMTPARATYPE) of LOOKS))
(\ATMOUT FILE (fetch (FMTSPEC FMTPARASUBTYPE) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTSTYLE) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTCHARSTYLES) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTNEWPAGEBEFORE) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTNEWPAGEAFTER) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTHEADINGKEEP) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTKEEP) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTBASETOBASE) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTREVISED) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTCOLUMN) of LOOKS))
(\ARBOUT FILE (fetch (FMTSPEC FMTCHARSTYLES) of LOOKS))

```

;;; Now go fill in the length field at the front of the LOOKS. (ALL looks info should be written out BEFORE this comment.)

```

(SETQ LEN (IDIFFERENCE (GETFILEPTR FILE)
FILEPOS)) ; The length of this set of looks
(SETFILEPTR FILE FILEPOS) ; Go write the length field
(\SMALLPOUT FILE LEN)
(SETFILEPTR FILE -1) ; And back to the end of the file

```

```

])

```

```

)

```

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```

```

(GLOBALVARS TEDIT.INPUT.FORMATS *TEDIT-FILE-READTABLE*)
)

```

```

(RPAQ? TEDIT.INPUT.FORMATS NIL)

```

```

(RPAQ? *TEDIT-FILE-READTABLE* (COPYREADTABLE \ORIGREADTABLE))

```

;; For converting old incoming format. Cutover 5/22/85 to permit looks changes in the future.

```

(DEFINEQ

```

```

(TEDIT.BUILD.PCTB2

```

```

[LAMBDA (TEXT TEXTOBJ PCCOUNT START END DEFAULTLOOKS) ; Edited 4-May-93 16:27 by jds

```

```

;; READ OBSOLETE FORMATS OF TEDIT FILE
;; START = 1st char of file to read from, if specified
;; END = use this as eofptr of file. For use in reading files within files.

```

```

(PROG (SEL LINES PCTB PC OLDPC TYPECODE PCLEN CHLOOKSSEEN NEWPC PARALOOKSSEEN PIECEINFOCH# CACHE CACHE?
TTEXTOBJ USER.CMFILE TSTREAM USERFILEFORMAT USERTEMP EXISTINGCHARLOOKS EXLOOK EXISTINGFMTSPECS
(CURFILECH# (OR START 0))
(CURCH# 1)
(TEXTSTREAM (AND TEXTOBJ (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)))
LOOKSHASH PARAHASH) ; Get the number of pieces needed (if AN formatted file),
; otherwise PCCOUNT will be NIL

```

```

[SETQ DEFAULTPARALOOKS (OR DEFAULTPARALOOKS (COND
(TEXTOBJ (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
(T (create FMTSPEC using TEDIT.DEFAULT.FMTSPEC)

```

```

(SETQ DEFAULTLOOKS (OR DEFAULTLOOKS (CHARLOOKS.FROM.FONT DEFAULTFONT)))
(AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with NIL)) ; Start by assuming no page formatting

```

```

(SETQ PCTB (\MAKEPCTB NIL PCCOUNT))
(replace (TEXTOBJ PCTB) of TEXTOBJ with PCTB)
(SETFILEPTR TEXT (IDIFFERENCE (OR END (GETEOFPTR TEXT))
8))

```

```

(SETQ PIECEINFOCH# (\DWIN TEXT))
(SETFILEPTR TEXT PIECEINFOCH#)

```

```

(bind (OLDPC _ NIL)
(OLDPARALOOKS _ DEFAULTPARALOOKS) for I from 1 to PCCOUNT as PCN from \FirstPieceOffset
by \EltsPerPiece
do (SETQ PC NIL)

```

```

; This loop may not really read a piece, so we have to distinguish
; that case.

```

```

(SETQ PCLEN (\DWIN TEXT))
(SETQ TYPECODE (\SMALLPIN TEXT)) ; What kind of piece is it?

```

```

(SELECTC TYPECODE
(\PieceDescriptorPAGEFRAME ; This is page layout info for the file
(AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with (TEDIT.GET.PAGEFRAMES TEXT)
))

```

```

(add PCN (IMINUS \EltsPerPiece)) ; This didn't create a piece -- don't count it in the PCTB
; placement.

```

```

)
(\PieceDescriptorCHARLOOKSLIST ; This is the list of CHARLOOKSs used in this document.
(replace (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ with (\TEDIT.GET.CHARLOOKS.LIST2 TEXT))
; Read the list of looks used in this document.

```

```

[SETQ LOOKSHASH (ARRAY (FLENGTH (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ]
                                ; Build an array of the looks, so the reader can index them.
(for I from 1 as LOOKS in (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ)
  do (SETA LOOKSHASH I LOOKS))
(add PCN (IMINUS \EltsPerPiece)) ; This didn't create a piece -- don't count it in the PCTB
                                ; placement.
(add I -1))
(\PieceDescriptorPARALOOKSLIST ; This is the list of PARALOOKSs used in this document.
(replace (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ with (\TEDIT.GET.PARALOOKS.LIST2 TEXT))
                                ; Read the list of looks used in this document.
[SETQ PARAHASH (ARRAY (FLENGTH (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ]
                                ; Build an array of the looks, so the reader can index them.
(for I from 1 as LOOKS in (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ)
  do (SETA PARAHASH I LOOKS))
(add PCN (IMINUS \EltsPerPiece)) ; This didn't create a piece -- don't count it in the PCTB
                                ; placement.
(add I -1))
(\PieceDescriptorPARA ; Reading a new set of paragraph looks.
(AND OLDPC (replace (PIECE PPARALAST) of OLDPC with T))
                                ; Mark the end of the preceding paragraph.
(SETQ OLDPARALOOKS (\TEDIT.GET.PARALOOKS TEXT PARAHASH))
                                ; Get the new set of looks, for use by later pieces.
(AND TEXTOBJ (replace (TEXTOBJ FORMATTEDP) of TEXTOBJ with T))
                                ; Mark the document as containing paragraph formatting info
(add PCN (IMINUS \EltsPerPiece)) ; This didn't create a piece -- don't count it in the PCTB
                                ; placement.
)
(\PieceDescriptorLOOKS ; New character looks. Build a piece to describe those
                        ; characters.
(SETQ PC
  (create PIECE
    PFILE _ TEXT
    PFPOS _ CURFILECH#
    PLEN _ PCLEN
    PREVPIECE _ OLDPC
    PPARALOOKS _ OLDPARALOOKS)) ; Build the new piece
(\TEDIT.GET.CHARLOOKS PC TEXT LOOKSHASH OLDPC) ; Read the character looks for this guy.
(COND
  [OLDPC ; If there's a prior piece, hook this one on the chain.
    (replace (PIECE NEXTPIECE) of OLDPC with PC)
  (COND
    ((AND (fetch (PIECE PFATP) of PC)
      (NOT (fetch (PIECE PFATP) of OLDPC)))
      ; Switching from not-fat to fat. Add 3 bytes for the 255-255-0
      (add (fetch (PIECE PFPOS) of PC)
        3)
      (add CURFILECH# -3))
    ((AND (fetch (PIECE PFATP) of OLDPC)
      (NOT (fetch (PIECE PFATP) of PC)))
      ; Switching from fat to not-fat. Add 3 bytes for the 255-0
      (add (fetch (PIECE PFPOS) of PC)
        2)
      ((fetch (PIECE PFATP) of PC) ; Switching from not-fat to fat. Add 3 bytes for the 255-255-0
      (add (fetch (PIECE PFPOS) of PC)
        3)
      (add CURFILECH# -3)))
    (add CURFILECH# PCLEN) ; And note the passing of characters.
  )
)
(\PieceDescriptorOBJECT ; It describes an object. Read that, and perhaps some
                        ; description of the charlooks to go with it.
(SETQ PC
  (create PIECE
    PFILE _ TEXT
    PFPOS _ CURFILECH#
    PLEN _ PCLEN
    PREVPIECE _ OLDPC
    PPARALOOKS _ OLDPARALOOKS))
(COND
  [OLDPC ; If there's a prior piece, hook this one on the chain.
    (replace (PIECE NEXTPIECE) of OLDPC with PC))
(TEDIT.GET.OBJECT TEXTSTREAM PC TEXT CURFILECH#)
(add CURFILECH# (fetch (PIECE PLEN) of PC))
(COND
  ((NOT (ZEROP (\BIN TEXT))) ; There are new character looks for this object. Read them in.
    (replace (PIECE PLOOKS) of PC with (\TEDIT.GET.SINGLE.CHARLOOKS2 TEXT)))
  (T ; No new looks; steal them from the prior piece.
    (replace (PIECE PLOOKS) of PC with (OR (AND OLDPC (fetch (PIECE PLOOKS) of OLDPC))
      DEFAULTLOOKS))
    (replace (PIECE PLEN) of PC with 1) ; OBJECTs are officially one character long.
  )
)
(SHOULDNT "Impossible piece-type code in BUILD.PCTB"))
(COND
  [PC ; If we created a piece, save it in the table.

```

```

                (\INSERTPIECE PC 'LASTPIECE TEXTOBJ)
                (add CURCH# (fetch (PIECE PLEN) of PC))
                (SETQ OLDPC PC)))
finally
(RETURN PCTB))
(* INSERT-BRT (CREATEPCNODE CURCH#
(QUOTE LASTPIECE)) PCTB))

```

(\TEDIT.GET.CHARLOOKS.LIST2

```

[LAMBDA (FILE)
  (for I from 1 to (\SMALLPIN FILE) collect (\TEDIT.GET.SINGLE.CHARLOOKS2 FILE))
(* jds "22-May-85 14:28"
* Read the list of CHARLOOKSs from the file.)

```

(\TEDIT.GET.SINGLE.CHARLOOKS2

```

[LAMBDA (FILE)
  (PROG* ((LOOKS (create CHARLOOKS))
    FONT STR NAME NAMELEN SIZE SUPER SUB PROPS STYLESTR USERSTR)
    (SETQ NAME (\ARBIN FILE))
    (SETQ SIZE (\SMALLPIN FILE))
    (SETQ SUPER (\SMALLPIN FILE))
    (replace (CHARLOOKS CLSTYLE) of LOOKS with (OR (\ARBIN FILE)
    0))
    (replace (CHARLOOKS CLUSERINFO) of LOOKS with (\ARBIN FILE))
    (SETQ PROPS (\SMALLPIN FILE))
    (with CHARLOOKS LOOKS [SETQ CLLEADER (NOT (ZEROP (LOGAND 2048 PROPS)
    [SETQ CLINVERTED (NOT (ZEROP (LOGAND 1024 PROPS)
    [SETQ CLBOLD (NOT (ZEROP (LOGAND 512 PROPS)
    [SETQ CLITAL (NOT (ZEROP (LOGAND 256 PROPS)
    [SETQ CLULINE (NOT (ZEROP (LOGAND 128 PROPS)
    [SETQ CLOLINE (NOT (ZEROP (LOGAND 64 PROPS)
    [SETQ CLSTRIKE (NOT (ZEROP (LOGAND 32 PROPS)
    [SETQ CLSMALLCAP (NOT (ZEROP (LOGAND 16 PROPS)
    [SETQ CLPROTECTED (NOT (ZEROP (LOGAND 8 PROPS)
    [SETQ CLINVISIBLE (NOT (ZEROP (LOGAND 4 PROPS)
    [SETQ CLSELHERE (NOT (ZEROP (LOGAND 2 PROPS)
    [SETQ CLCANCOPY (NOT (ZEROP (LOGAND 1 PROPS)
    (SETQ CLSIZE SIZE)
    (SETQ CLOFFSET SUPER))
    (replace (CHARLOOKS CLFONT) of LOOKS with (COND
      ((LISTP NAME)
        (* This was a font class. Restore it.)
        (FONTCLASS (pop NAME)
          NAME))
      ((AND NAME (NOT (ZEROP SIZE)))
        (FONTCREATE NAME SIZE (COND
          ((AND (fetch (CHARLOOKS CLBOLD)
            of LOOKS)
            (fetch (CHARLOOKS CLITAL)
              of LOOKS))
            'BOLDITALIC)
          ((fetch (CHARLOOKS CLBOLD)
            of LOOKS)
            'BOLD)
          ((fetch (CHARLOOKS CLITAL)
            of LOOKS)
            'ITALIC]
      )
    (RETURN LOOKS]))
; Edited 30-May-91 20:26 by jds
(* Read a set of CHARLOOKS from FILE)

```

(\TEDIT.PUT.SINGLE.PARALOOKS2

```

[LAMBDA (FILE LOOKS)
  (PROG (DEFAULTTAB TABSPECS OUTPUTFORMAT LEN)
    (\SMALLPOUT FILE (fetch (FMTSPEC 1STLEFTMAR) of LOOKS))
    (\SMALLPOUT FILE (fetch (FMTSPEC LEFTMAR) of LOOKS))
    (\SMALLPOUT FILE (fetch (FMTSPEC RIGHTMAR) of LOOKS))
    (\SMALLPOUT FILE (fetch (FMTSPEC LEADBEFORE) of LOOKS))
    (\SMALLPOUT FILE (fetch (FMTSPEC LEADAFTER) of LOOKS))
    (\SMALLPOUT FILE (fetch (FMTSPEC LINELEAD) of LOOKS))
    (SETQ DEFAULTTAB (CAR (fetch (FMTSPEC TABSPEC) of LOOKS)))
    (SETQ TABSPECS (CDR (fetch (FMTSPEC TABSPEC) of LOOKS)))
    (COND
      ((AND (fetch (FMTSPEC TABSPEC) of LOOKS)
        (OR DEFAULTTAB TABSPECS))
        (* There are tab specs to save, or there is a default tab setting to save)

        (\BOUT FILE 3))
      (T
        (\BOUT FILE 2)))
    (\BOUT FILE (SELECTQ (fetch (FMTSPEC QUAD) of LOOKS)
      (LEFT 1)
      (RIGHT 2)
      ((CENTER CENTERED)
        (* There are no tab looks. Just let him go.)

```



```

        3)
        ((JUST JUSTIFIED)
        4)
        (SHOULDNT)))
[COND
  ((OR TABSPECS DEFAULTTAB)
   (COND
    (DEFAULTTAB (\SMALLPOUT FILE DEFAULTTAB))
    (T (\SMALLPOUT FILE 0)))
   (\BOUT FILE (LENGTH TABSPECS))
   (COND
    (TABSPECS
     (for TAB in TABSPECS do (\SMALLPOUT FILE
                               (SELECTQ (fetch TABX of TAB)
                                         (* And setting.)
                                         (\BOUT FILE (SELECTQ (fetch TABKIND of TAB)
                                                             (LEFT 0)
                                                             (RIGHT 1)
                                                             (CENTERED 2)
                                                             (DECIMAL 3)
                                                             (SHOULDNT))))
                               (* Tab type)
                               ]
     )
    )
   (\SMALLPOUT FILE (OR (fetch (FMTSPEC FMTSPECIALX) of LOOKS)
                        0))
   (\SMALLPOUT FILE (OR (fetch (FMTSPEC FMTSPECIALY) of LOOKS)
                        0))
   (\ARBOUT FILE (fetch (FMTSPEC FMTUSERINFO) of LOOKS))
   (\ATMOUT FILE (fetch (FMTSPEC FMTPARATYPE) of LOOKS))
   (\ATMOUT FILE (fetch (FMTSPEC FMTPARASUBTYPE) of LOOKS))
   (\ARBOUT FILE (fetch (FMTSPEC FMTSTYLE) of LOOKS))
   (\ARBOUT FILE (fetch (FMTSPEC FMTCHARSTYLES) of LOOKS))
   (\ARBOUT FILE (fetch (FMTSPEC FMTNEWPAGEBEFORE) of LOOKS))
   (\ARBOUT FILE (fetch (FMTSPEC FMTNEWPAGEAFTER) of LOOKS]))

```

(\TEDIT.PUT.SINGLE.CHARLOOKS2

[LAMBDA (FILE LOOKS)

; Edited 30-May-91 20:26 by jds

(* Put out a single CHARLOOKS description.)

```

(PROG ((FONT (fetch (CHARLOOKS CLFONT) of LOOKS))
      STR LEN)
 [COND
  ((type? FONTCLASS FONT)
   (\ARBOUT FILE (FONTCLASSUNPARSE FONT)))
  (T
   (\ATMOUT FILE (FONTPROP FONT 'FAMILY)
   (\SMALLPOUT FILE (OR (FONTPROP FONT 'SIZE)
                       0))
   (\SMALLPOUT FILE (OR (fetch (CHARLOOKS CLOFFSET) of LOOKS)
                       0))
   (COND
    ([AND (fetch (CHARLOOKS CLSTYLE) of LOOKS)
          (NOT (ZEROP (fetch (CHARLOOKS CLSTYLE) of LOOKS))
           (\ARBOUT FILE (fetch (CHARLOOKS CLSTYLE) of LOOKS)))
    (T (\SMALLPOUT FILE 0)))
   (COND
    ((fetch (CHARLOOKS CLUSERINFO) of LOOKS)
     (\ARBOUT FILE (fetch (CHARLOOKS CLUSERINFO) of LOOKS)))
    (T (\SMALLPOUT FILE 0)))
   (\SMALLPOUT FILE (LOGOR (COND
                           ((fetch (CHARLOOKS CLLEADER) of LOOKS)
                            (* Dotted-leader; relevant only to TABs)
                            2048)
                           (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLINVERTED) of LOOKS)
                             (* Inverse-video)
                             1024)
                            (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLBOLD) of LOOKS)
                             512)
                            (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLITAL) of LOOKS)
                             256)
                            (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLULINE) of LOOKS)
                             128)
                            (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLOLINE) of LOOKS)
                             64)
                            (T 0))
                           (COND
                            ((fetch (CHARLOOKS CLSTRIKE) of LOOKS)
                             32)

```

```

      (T 0))
    (COND
      ((fetch (CHARLOOKS CLSMALLCAP) of LOOKS)
        16)
      (T 0))
    (COND
      ((fetch (CHARLOOKS CLPROTECTED) of LOOKS)
        8)
      (T 0))
    (COND
      ((fetch (CHARLOOKS CLINVISIBLE) of LOOKS)
        NIL 4)
      (T 0))
    (COND
      ((fetch (CHARLOOKS CLSELHERE) of LOOKS)
        2)
      (T 0))
    (COND
      ((fetch (CHARLOOKS CLCANCOPY) of LOOKS)
        1)
      (T 0))

```

(\TEDIT.GET.PARALOOKS.LIST2

```

[LAMBDA (FILE)
  (* jds "22-May-85 14:28"
    (* Read the list of CHARLOOKSs from the file.)
  (for I from 1 to (\SMALLPIN FILE) collect (\TEDIT.GET.SINGLE.PARALOOKS2 FILE])

```

(\TEDIT.GET.SINGLE.PARALOOKS2

```

[LAMBDA (FILE)
  (* Edited 30-May-91 20:33 by jds
    (* Read a paragraph format spec from the FILE, and return it for
  later use.)
  (PROG ((LOOKS (create FMTSPEC))
    TABFLG DEFAULTTAB TABCOUNT TABS TABSPEC)
    (replace (FMTSPEC 1STLEFTMAR) of LOOKS with (\SMALLPIN FILE))
      (* Left margin for the first line of the paragraph)
    (replace (FMTSPEC LEFTMAR) of LOOKS with (\SMALLPIN FILE))
      (* Left margin for the rest of the paragraph)
    (replace (FMTSPEC RIGHTMAR) of LOOKS with (\SMALLPIN FILE))
      (* Right margin for the paragraph)
    (replace (FMTSPEC LEADBEFORE) of LOOKS with (\SMALLPIN FILE))
      (* Leading before the paragraph)
    (replace (FMTSPEC LEADAFTER) of LOOKS with (\SMALLPIN FILE))
      (* Lead after the paragraph)
    (replace (FMTSPEC LINELEAD) of LOOKS with (\SMALLPIN FILE))
      (* inter-line leading)
    (replace (FMTSPEC TABSPEC) of LOOKS with (SETQ TABSPEC (CONS NIL NIL)))
      (* Will be tab specs)
    (SETQ TABFLG (\BIN FILE))
    (replace (FMTSPEC QUAD) of LOOKS with (SELECTC (\BIN FILE)
      (1 'LEFT)
      (2 'RIGHT)
      (3 'CENTERED)
      (4 'JUSTIFIED)
      (SHOULDNT)))
    (COND
      ((NOT (ZEROP (LOGAND TABFLG 1)))
        (* There are tabs to read)
        (SETQ DEFAULTTAB (\SMALLPIN FILE))
        (SETQ TABCOUNT (\BIN FILE))
        [SETQ TABS (for TAB# from 1 to TABCOUNT collect (create TAB
          TABX _ (\SMALLPIN FILE)
          TABKIND _ (SELECTQ (\BIN FILE)
            (0 'LEFT)
            (1 'RIGHT)
            (2 'CENTERED)
            (3 'DECIMAL)
            (SHOULDNT))
          (OR (ZEROP DEFAULTTAB)
            (RPLACA TABSPEC DEFAULTTAB))
            (RPLACD TABSPEC TABS)))
        (COND
          ((NOT (ZEROP (LOGAND TABFLG 2)))
            (* There are other paragraph parameters to be read.)
            (replace (FMTSPEC FMTSPECIALX) of LOOKS with (\SMALLPIN FILE))
              (* Special X location on page for this paragraph)
            (replace (FMTSPEC FMTSPECIALY) of LOOKS with (\SMALLPIN FILE))
            (replace (FMTSPEC FMTUSERINFO) of LOOKS with (\ARBIN FILE))
            (replace (FMTSPEC FMTPARATYPE) of LOOKS with (\ATMIN FILE))
            (replace (FMTSPEC FMTPARASUBTYPE) of LOOKS with (\ATMIN FILE))
            (replace (FMTSPEC FMTSTYLE) of LOOKS with (\ARBIN FILE))
            (replace (FMTSPEC FMTCHARSTYLES) of LOOKS with (\ARBIN FILE))
            (replace (FMTSPEC FMTNEWPAGEBEFORE) of LOOKS with (\ARBIN FILE))
            (replace (FMTSPEC FMTNEWPAGEAFTER) of LOOKS with (\ARBIN FILE))
            (RETURN LOOKS])

```

(\TEDIT.PUT.PCTB2

```

[LAMBDA (TEXTOBJ OFILE UNFORMATTED? SEPARATEFORMAT) ; Edited 30-May-91 20:24 by jds
;; Put a representation of the piece table onto OFILE, preserving font changes and paragraph looks. UNFORMATTED? means write no font or
;; formatting info.
(PROG (OCURSOR CH PC PFILE PSTR POBJ OFILELEN OLDLOOKS (OLDPARALOOKS (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
      OLDCH# CURCH# PREVPC (FONTFILE NIL)
      (PCCOUNT 0)
      TRUEFILE CHARLOOKSLST PARALOOKSLST (TEDIT.PUT.FINISHEDFORMS NIL)
      (EDITSTENTATIVE (TEXTPROP TEXTOBJ 'TEDIT.TENTATIVE))
      (PARALOOKSSEEN NIL)
      (FORMATTINGLEVEL (TEDIT.FORMATTEDFILEP TEXTOBJ))
      (CACHE (TEXTPROP TEXTOBJ 'CACHE))
      CH#S PREVFATP PREVPREVPC LOOKSHASH PARAHASH)
  (SETQ PC (\EDITELT (fetch (TEXTOBJ PCTB) of TEXTOBJ)
                    (ADD1 \FirstPieceOffset))) ; First piece in the document
  (SETQ OLDLOOKS (OR (AND (type? PIECE PC)
                        (fetch (PIECE PLOOKS) of PC))
                    (fetch (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ)
                    TEDIT.DEFAULT.CHARLOOKS)) ; Starting looks
  (COND
    ((NEQ (fetch (STREAM EOLCONVENTION) of OFILE)
          CR.EOLC) ; This file is on a non-CR host; make a note to cache it
     (SETQ TRUEFILE OFILE) ; Remember where the file should wind up.
     (SETQ OFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)) ; And open a temp file to write it to.
    ))
  [SETQ CURCH# (SETQ OLDCH# (ADD1 (GETFILEPTR OFILE)
  (COND
    ((fetch (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ) ; There is layout info for this file. Save it
     (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW))
     (TEDIT.PUT.PAGEFRAMES FONTFILE (fetch (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ))
     (add PCCOUNT 1)))
    (\TEDIT.FLUSH.UNUSED.LOOKS TEXTOBJ PC) ; Run thru the lists of char & para looks and remove any that
                                           ; aren't in use
  (COND
    ([AND (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ)
          (OR (IGREATERP (FLENGTH (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
                        1)
              (NOT (EQFMTSPEC (CAR (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))
                             TEDIT.DEFAULT.FMTSPEC))
          ))
    ;; There are paragraph looks in this document that don't match the default -- save the list of them for later retrieval.
    [OR FONTFILE (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)
                  ; Create the font-info file if it doesn't exist yet
                  (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ))]
    (SETQ PARAHASH (\TEDIT.PUT.PARALOOKS.LIST2 FONTFILE (fetch (TEXTOBJ TXTPARALOOKSLIST) of TEXTOBJ)))
    (SETQ PARALOOKSSEEN T)))
  [COND
    ((OR PARALOOKSSEEN FORMATTINGLEVEL)
     ;; There are character looks in this document that don't match the default (or paragraph formatting, which forces looks to be saved) --
     ;; save the list for later retrieval.
     [OR FONTFILE (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)
                     (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST2 FONTFILE (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ)
                     (while PC do (COND
                        ([AND (NOT (ZEROP (fetch (PIECE PLEN) of PC)))
                              (OR (NOT PREVPC)
                                  (fetch (PIECE PPARALAST) of PREVPC))
                              (OR PARALOOKSSEEN (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
                                                                    (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ)
                              ))
                        [OR FONTFILE (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)
                                                  ; The last piece ended a paragraph, so send out new para looks
                                                  ; Create the formatting-info file, if it didn't exist before.
                                                  (COND
                                ((NEQ CURCH# OLDCH#) ; There were prior characters that hadn't been described in a
                                                         ; piece yet. Describe them
                                [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE
                                                                    (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ)
                                                                    (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDLOOKS PREVP PC EDITSTENTATIVE
                                                                    LOOKSHASH PREVFATP)
                                (add PCCOUNT 1)
                                (SETQ OLDCH# CURCH#) ; And now we've described all the characters up to the current
                                                         ; one.
                                ))
                                (\TEDIT.PUT.PARALOOKS FONTFILE PC PARAHASH)
                                (SETQ PARALOOKSSEEN T) ; Remember that we've seen a foreign paraloooks, and must
                                                         ; henceforth note para boundaries
                                (add PCCOUNT 1)))
                                (COND
                                  [(fetch (PIECE POBJ) of PC) ; It's an object -- go use its PUTFN
                                   [OR FONTFILE (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)
                                                         ; Create the font-info file, if need be.
                                   (COND
                                    ((AND (NEQ CURCH# OLDCH#)
                                             PREVPC) ; There were prior characters that hadn't been described in a
                                                         ; piece yet. Describe them
                                    [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE

```

```

                                (fetch (TEXTOBJ TXTCHARLOOKSLIST) of TEXTOBJ]
(\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDLOOKS PREVPC EDITSTENTATIVE
 LOOKSHASH PREVFPATP)
(add PCCOUNT 1)
(SETQ OLDCH# CURCH#)                                ; And now we've described all the characters up to the current
                                                    ; one.

) )
; If the prior thing was text, send along its descriptor.
(add CURCH# (TEDIT.PUT.OBJECT PC OFILE FONTFILE CURCH#))
; Send out the object
(add PCCOUNT 1)
(SETQ OLDCH# CURCH#)
(COND
  ([OR (NOT (EQCLOOKS OLDLOOKS (fetch (PIECE PLOOKS) of PC)))
    (NEQ (fetch (PIECE PFATP) of PC)
      (AND PREVPC (fetch (PIECE PFATP) of PREVPC)))
    [AND EDITSTENTATIVE (NEQ (fetch (PIECE PNEW) of PC)
      (AND PREVPC (fetch (PIECE PNEW) of PREVPC)
        (AND (OR (NOT PREVPC)
          (fetch (PIECE PPARALAST) of PREVPC))
          (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
            (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ]
            ; The OBJECT has different ooks from before
          (\BOUT FONTFILE 1)
          (\TEDIT.PUT.SINGLE.CHARLOOKS FONTFILE (fetch (PIECE PLOOKS) of PC))
          (SETQ OLDLOOKS (fetch (PIECE PLOOKS) of PC)))
        (T
          ; No differences. Don't write any charlooks, and mark that fact
          (\BOUT FONTFILE 0) ; MAKE BLOODY SURE THAT THE NEXT RUN OF
          ; CHARACTERS GETS ITS OWN LOOKS
        )
      ]
    )
    ; It's not an object.
  ]
  (COND
    ([OR (NOT (EQCLOOKS OLDLOOKS (fetch (PIECE PLOOKS) of PC)))
      (NEQ (fetch (PIECE PFATP) of PC)
        (AND PREVPC (fetch (PIECE PFATP) of PREVPC)))
      [AND EDITSTENTATIVE (NEQ (fetch (PIECE PNEW) of PC)
        (AND PREVPC (fetch (PIECE PNEW) of PREVPC)
          (AND (OR (NOT PREVPC)
            (fetch (PIECE PPARALAST) of PREVPC))
            (NOT (EQFMTSPEC (fetch (PIECE PPARALOOKS) of PC)
              (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ]
              ; We have a piece with new looks.
            [OR FONTFILE (SETQ FONTFILE (OPENFILE '{NODIRCORE} 'BOTH 'NEW)
              (COND
                ((NOT (IEQP OLDCH# CURCH#)) ; If there were looks past, and if the run was not empty, save a
                ; piece for its looks
                [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE
                  (fetch (TEXTOBJ TXTCHARLOOKSLIST)
                    of TEXTOBJ]
                  (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDLOOKS PREVPC EDITSTENTATIVE
                    LOOKSHASH PREVFPATP)
                  (add PCCOUNT 1)))
                (SETQ OLDLOOKS (fetch (PIECE PLOOKS) of PC))
                (SETQ OLDCH# CURCH#)
                (COND
                  [PREVFPATP (COND
                    ((fetch (PIECE PFATP) of PC))
                    (T
                      ; Switching from FAT to thin
                      (BOUT OFILE 255)
                      (BOUT OFILE 0)
                      (add CURCH# 2]
                    )
                  )
                  ((fetch (PIECE PFATP) of PC) ; Switching from thin to fat
                    (BOUT OFILE 255)
                    (BOUT OFILE 255)
                    (BOUT OFILE 0)
                    (add CURCH# 3)))
                  (SETQ PREVFPATP (fetch (PIECE PFATP) of PC]
                    ; Now dump out the non-object contents of the piece.
                  (COND
                    [(SETQ PFILE (fetch (PIECE PFILE) of PC))
                      ; It's on a file. Copy it.
                    ]
                    [OR (OPENP PFILE)
                      (replace (PIECE PFILE) of PC with (SETQ PFILE (OPENSTREAM (fetch (STREAM
                        FULLNAME
                        )
                        of PFILE)
                        'INPUT]
                      ; Make sure the file is open.
                    )
                    (COPYBYTES PFILE OFILE (fetch (PIECE PFPOS) of PC)
                      (IPLUS (fetch (PIECE PFPOS) of PC)
                        (COND
                          ((fetch (PIECE PFATP) of PC)
                            ; For fat file pieces, copy twice as many bytes as characters.
                          (UNFOLD (fetch (PIECE PLEN) of PC)
                            2))
                          (T (fetch (PIECE PLEN) of PC]
                        )
                      )
                    )

```

```

((SETQ PSTR (fetch (PIECE PSTR) of PC)) ; It's in a string. Just print it.
(COND
  [(fetch (PIECE PFATP) of PC) ; The string is fat: Copy twice as many bytes as chars.
    (for I from 1 to (fetch (PIECE PLEN) of PC) as CH instring PSTR
      do (\BOUT OFILE (\CHARSET CH)
          (\BOUT OFILE (\CHAR8CODE CH)
            (T
              (for I from 1 to (fetch (PIECE PLEN) of PC) as CH instring PSTR
                do (\BOUT OFILE CH)
              )
            )
          )
        )
      )
    ]
  [COND
    ((AND (NOT CACHE)
          (RANDACCESSP OFILE))
      ; CSLI leave the pieces and the pctb alone and just write the file
      ; if its cached or not randomaccess
      (push CH#S (SUB1 CURCH#)
        )
      )
    [COND
      ((fetch (PIECE PFATP) of PC)
        (add CURCH# (UNFOLD (fetch (PIECE PLEN) of PC)
          2)))
      (T (add CURCH# (fetch (PIECE PLEN) of PC)
        )
      )
    ]
    ; Keep running track of where in the file we are.
  )
  )
  (SETQ PREVPREVPVC PREVPVC)
  (SETQ PREVPVC PC)
  (SETQ PC (fetch (PIECE NEXTPIECE) of PC))
  finally ; Put out a piece describing the last characters in the file.
  (COND
    ((AND FONTFILE (NEQ OLDCH# CURCH#))
      ; Only if there WERE characters, and only if there's a need for
      ; font information
      [OR LOOKSHASH (SETQ LOOKSHASH (\TEDIT.PUT.CHARLOOKS.LIST FONTFILE (fetch (TEXTOBJ
        TXCHARLOOKSLIST
        )
        of TEXTOBJ)
      )
      (\TEDIT.PUT.CHARLOOKS FONTFILE OLDCH# CURCH# OLDLOOKS PREVPVC EDITSTENTATIVE LOOKSHASH
        PREVPREVPVC)
      ; Put out a description of the characters
      (add PCCOUNT 1))
      )
    (COND
      ((AND PARALOOKSSEEN (fetch (PIECE PPARALAST) of PREVPVC))
        ; The last piece contained the end of a paragraph. Make sure it
        ; gets noted.
        (\TEDIT.PUT.PARALOOKS FONTFILE PREVPVC PARASHASH)
        (add PCCOUNT 1]
      )
    )
  )
  (for FORM in TEDIT.PUT.FINISHEDFORMS do (EVAL FORM)) ; Do any user-specific cleanup
  (COND
    (TRUEFILE
      (COND
        ((AND FONTFILE (NOT UNFORMATTED?)
          (NOT SEPARATEFORMAT))
          ; Formatted file: Copy without converting.
          (COPYBYTES OFILE TRUEFILE 0 -1))
          ; Go ahead and convert the EOLCONVENTION, this is a
          ; plain-text file
          (COPYCHARS OFILE TRUEFILE 0 -1)))
        (SETQ OFILE TRUEFILE)))
    )
  [COND
    ((AND (OPENP OFILE)
      FONTFILE)
      ; We need to write format info.
      ; So remember the end of the plain-text part of the file
      ; # OF PIECES WE' NEED TO RECONSTRUCT THIS FILE
      ; Now the password for NEW format files: 31416
      (\DWOUT FONTFILE (GETEOFPTR OFILE))
      (\SMALLPOUT FONTFILE PCCOUNT)
      (\SMALLPOUT FONTFILE 31417)
      (COND
        ((AND (NOT UNFORMATTED?)
          (NOT SEPARATEFORMAT))
          ;; Only write fmtg info at the end if we want it there--not if we want plain text or want it kept separate.
          (COPYBYTES FONTFILE OFILE 0 (GETEOFPTR FONTFILE))
          ; Copy the font information to the file trailer
          )
        )
      )
    (T))
    (CLOSEF FONTFILE)
    (COND
      ((NOT SEPARATEFORMAT)
        ; Unless we want the formatting info separately, delete the file
        ; (since FONTFILE is a stream, we should not need to delete it at
        ; all) (DELFILE FONTFILE)
      )
    )
  ]
  (replace (TEXTOBJ DEFAULTCHARLOOKS) of TEXTOBJ with (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ
    DEFAULTCHARLOOKS
    )
    of TEXTOBJ)
    TEXTOBJ))
    ; Re-add the default and caret looks's to the lists, since they may
    ; not have been really saved.
  (replace (TEXTOBJ CARETLOOKS) of TEXTOBJ with (\TEDIT.UNIQIFY.CHARLOOKS (fetch (TEXTOBJ CARETLOOKS)
    of TEXTOBJ)
    TEXTOBJ))

```

```

(replace (TEXTOBJ FMTSPEC) of TEXTOBJ with (\TEDIT.UNIQIFY.PARALOOKS (fetch (TEXTOBJ FMTSPEC)
                                                                    of TEXTOBJ)
                                                                    TEXTOBJ))

(RETURN (CONS (COND
                (UNFORMATTED? NIL)
                (T FONTFILE))
            CH#S])

```

(TEDIT.PUT.CHARLOOKS.LIST2

[LAMBDA (FILE LOOKSLIST)

(* jds "22-May-85 15:12")

(* Write the list of CHARLOOKSs into the font file.)

(* Returns a hasharray that will map from a given CHARLOOKS to that CHARLOOKS' position in the list we wrote on the file. Those position numbers are then written in the individual looks descriptions, and are used to reconstruct the piece looks when the file is read back in.)

```

(PROG ((LOOKSHASH (HASHARRAY 50)))
  (\DWOUT FILE 0)
  (\SMALLPOUT FILE \PieceDescriptorCHARLOOKSLIST)
  (\SMALLPOUT FILE (LENGTH LOOKSLIST))
  (for I from 1 as LOOKS in LOOKSLIST do
    (\TEDIT.PUT.SINGLE.CHARLOOKS2 FILE LOOKS)
    (PUTHASH LOOKS I LOOKSHASH)
    (* No characters are described by this pseudo-piece entry.)
    (* Mark it as containing the list of CHARLOOKSs)
    (* How many CHARLOOKSs there are in the list)
    (* Write each charlooks, in the order they appear in the list.)
    (* Write out the description)
    (* And save it in the hash table so people can find its index.))

  (RETURN LOOKSHASH))

```

(TEDIT.PUT.PARALOOKS.LIST2

[LAMBDA (FILE LOOKSLIST)

(* jds "22-May-85 15:09")

(* Write the list of FMTSPECS into the font file.)

```

(PROG ((LOOKSHASH (HASHARRAY 50)))
  (\DWOUT FILE 0)
  (\SMALLPOUT FILE \PieceDescriptorPARALOOKSLIST)
  (\SMALLPOUT FILE (LENGTH LOOKSLIST))
  (for I from 1 as LOOKS in LOOKSLIST do (\TEDIT.PUT.SINGLE.PARALOOKS2 FILE LOOKS)
    (* Write out the description)
    (PUTHASH LOOKS I LOOKSHASH)
    (* And save it in the hash table so people can find its index.))

  (RETURN LOOKSHASH))

```

)

;; For converting incoming old-format files (1/27/85 cutover)

(DEFINEQ

(TEDIT.BUILD.PCTB1

[LAMBDA (TEXT TEXTOBJ PCCOUNT START END DEFAULTLOOKS)

; Edited 22-May-92 18:00 by jds

;;; READ OBSOLETE FORMATS OF TEDIT FILE

;; START = 1st char of file to read from, if specified

;; END = use this as eofptr of file. For use in reading files within files.

```

(PROG [SEL LINES PCTB PC OLDPC TYPECODE PCLEN CHLOOKSSEEN NEWPC PARALOOKSSEEN PIECEINFOCH# CACHE CACHE#
  TTEXTOBJ USER.CMFILE TSTREAM USERFILEFORMAT USERTEMP EXISTINGCHARLOOKS EXLOOK EXISTINGFMTSPECS
  (CURFILECH# (OR START 0))
  (CURCH# 1)
  (TEXTSTREAM (AND TEXTOBJ (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)]

```

;; Get the number of pieces needed (if AN formatted file), otherwise PCCOUNT will be NIL

```

[SETQ DEFAULTPARALOOKS (OR DEFAULTPARALOOKS (COND
  (TEXTOBJ (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
  (T (create FMTSPEC using TEDIT.DEFAULT.FMTSPEC))
  (SETQ DEFAULTLOOKS (OR DEFAULTLOOKS (CHARLOOKS.FROM.FONT DEFAULTFONT)))
  (AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with NIL))

```

; Start by assuming no page formatting

```

(SETQ PCTB (\MAKEPCTB NIL PCCOUNT))
(SETFILEPTR TEXT (IDIFFERENCE (OR END (GETEOFPTR TEXT))
  8))

```

(SETQ PIECEINFOCH# (\DWIN TEXT))

(SETFILEPTR TEXT PIECEINFOCH#)

(bind (OLDPC _ NIL)

(OLDPARALOOKS _ DEFAULTPARALOOKS) for I from 1 to PCCOUNT as PCN from \FirstPieceOffset

by \EltsPerPiece

do (SETQ PC NIL)

; This loop may not really read a piece, so we have to distinguish; that case.

(SETQ PCLEN (\DWIN TEXT))

(SETQ TYPECODE (\SMALLPIN TEXT))

; What kind of piece is it?

(SELECTC TYPECODE

(\PieceDescriptorPAGEFRAME

; This is page layout info for the file

(AND TEXTOBJ (replace (TEXTOBJ TXTPAGEFRAMES) of TEXTOBJ with (TEDIT.GET.PAGEFRAMES1 TEXT)))

(add PCN (IMINUS \EltsPerPiece))

; This didn't create a piece -- don't count it in the PCTB; placement.

```

)
(\PieceDescriptorPARA
  (AND OLDPC (replace (PIECE PPARALAST) of OLDPC with T)) ; Reading a new set of paragraph looks.
  (SETQ OLDPARALOOKS (\TEDIT.GET.PARALOOKS1 TEXT)) ; Mark the end of the preceding paragraph.
  (AND TEXTOBJ (replace (TEXTOBJ FORMATTEDP) of TEXTOBJ with T)) ; Get the new set of looks, for use by later pieces.
  (add PCN (IMINUS \EltPerPiece)) ; Mark the document as containing paragraph formatting info
  ; This didn't create a piece -- don't count it in the PCTB
  ; placement.
)

(\PieceDescriptorLOOKS
  ; New character looks. Build a piece to describe those
  ; characters.
  (SETQ PC
    (create PIECE
      PFILE _ TEXT
      PFPOS _ CURFILECH#
      PLEN _ PCLEN
      PREVPIECE _ OLDPC
      PPARALOOKS _ OLDPARALOOKS)) ; Build the new piece
  (COND
    (OLDPC ; If there's a prior piece, hook this one on the chain.
      (replace (PIECE NEXTPIECE) of OLDPC with PC)))
  (\TEDIT.GET.CHARLOOKS1 PC TEXT) ; Read the character looks for this guy.
  (add CURFILECH# (fetch (PIECE PLEN) of PC)) ; And note the passing of characters.
  )

(\PieceDescriptorOBJECT
  ; It describes an object. Read that, and perhaps some
  ; description of the charlooks to go with it.
  (SETQ PC
    (create PIECE
      PFILE _ TEXT
      PFPOS _ CURFILECH#
      PLEN _ PCLEN
      PREVPIECE _ OLDPC
      PPARALOOKS _ OLDPARALOOKS))
  (COND
    (OLDPC ; If there's a prior piece, hook this one on the chain.
      (replace (PIECE NEXTPIECE) of OLDPC with PC)))
  (\TEDIT.GET.OBJECT1 TEXTSTREAM PC TEXT CURFILECH#)
  (add CURFILECH# (fetch (PIECE PLEN) of PC))
  [COND
    ((NOT (ZEROP (\BIN TEXT))) ; There are new character looks for this object. Read them in.
      (\DWIN TEXT)
      (\WIN TEXT) ; Skip over the piece-type code we know has to be here.
      (\TEDIT.GET.CHARLOOKS1 PC TEXT))
    (T ; No new looks; steal them from the prior piece.
      (replace (PIECE PLOOKS) of PC with (OR (AND OLDPC (fetch (PIECE PLOOKS) of OLDPC))
        DEFAULTLOOKS))
      (replace (PIECE PLEN) of PC with 1) ; OBJECTs are officially one character long.
    )
  )
  (SHOULDNT "Impossible piece-type code in BUILD.PCTB"))
(COND
  (PC ; If we created a piece, save it in the table.
    [COND
      ((SETQ EXLOOK (for LOOK in EXISTINGCHARLOOKS thereis (EQCLOOKS (fetch (PIECE PLOOKS)
        of PC)
        LOOK))) ; These charlooks are a duplicate of pre-existing ones. Re-use
        ; the old one.
      (replace (PIECE PLOOKS) of PC with EXLOOK))
      (T (push EXISTINGCHARLOOKS (fetch (PIECE PLOOKS) of PC)]
    )
  )
  [COND
    ((SETQ EXLOOK (for LOOK in EXISTINGFMTSPECS thereis (EQFMTSPEC (fetch (PIECE PPARALOOKS)
      of PC)
      LOOK))) ; These paraloooks are a duplicate of pre-existing ones. Re-use
      ; the old one.
    (replace (PIECE PPARALOOKS) of PC with EXLOOK))
    (T (push EXISTINGFMTSPECS (fetch (PIECE PPARALOOKS) of PC)]
  )
  (INSERT-BRT (CREATEPCNODE CURCH# PC)
    PCTB)
  (add CURCH# (fetch (PIECE PLEN) of PC))
  (SETQ OLDPC PC))) finally (INSERT-BRT (CREATEPCNODE CURCH# 'LASTPIECE)
    PCTB))
(RETURN PCTB)]

```

(TEDIT.GET.PAGEFRAMES1

[LAMBDA (FILE)

(* jds " 1-Feb-85 14:55")

(* Read a bunch of page frames from the file, and return it.)

(TEDIT.PARSE.PAGEFRAMES1 (READ FILE))

(\TEDIT.GET.CHARLOOKS1

[LAMBDA (PC FILE)

; Edited 30-May-91 20:26 by jds

(* Read a description of PC's CHARLOOKS from FILE.)

```

(PROG (FONT STR NAME NAMELEN SIZE SUPER SUB PROPS STYLESTR USERSTR (LOOKS (create CHARLOOKS)))
  (replace (PIECE PLOOKS) of PC with LOOKS)
  (SETQ NAME (\ARBIN FILE))
  (SETQ SIZE (\SMALLPIN FILE))
  (SETQ SUPER (\SMALLPIN FILE))
  (SETQ SUB (\SMALLPIN FILE))
  (OR (ZEROP SUB)
    (SETQ SUPER (IMINUS SUB)))

```

(* The font name)

(* Size of the type, in points)

(* Superscripting distance)

(* former Subscripting distance)

(* If this is an old file, it'll have a subscript value not zero. Let those past and do the right thing.)

(COND

((NOT (ZEROP (\BIN FILE)))

(* This text is NEW. Mark it so.)

(replace (PIECE PNEW) of PC with T)))

[COND

((NOT (ZEROP (\BIN FILE)))

(* There is style or user information to be read)

(replace (CHARLOOKS CLSTYLE) of LOOKS with (OR (\ARBIN FILE)

0))

(replace (CHARLOOKS CLUSERINFO) of LOOKS with (\ARBIN FILE]

(SETQ PROPS (\SMALLPIN FILE))

(with CHARLOOKS LOOKS [SETQ CLBOLD (NOT (ZEROP (LOGAND 512 PROPS]

[SETQ CLITAL (NOT (ZEROP (LOGAND 256 PROPS]

[SETQ CLULINE (NOT (ZEROP (LOGAND 128 PROPS]

[SETQ CLOLINE (NOT (ZEROP (LOGAND 64 PROPS]

[SETQ CLSTRIKE (NOT (ZEROP (LOGAND 32 PROPS]

[SETQ CLSMALLCAP (NOT (ZEROP (LOGAND 16 PROPS]

[SETQ CLPROTECTED (NOT (ZEROP (LOGAND 8 PROPS]

[SETQ CLINVISIBLE (NOT (ZEROP (LOGAND 4 PROPS]

[SETQ CLSELHERE (NOT (ZEROP (LOGAND 2 PROPS]

[SETQ CLCANCOPY (NOT (ZEROP (LOGAND 1 PROPS]

(SETQ CLSIZE SIZE)

(SETQ CLOFFSET SUPER))

(replace (CHARLOOKS CLFONT) of LOOKS with (COND

((LISTP NAME)

(* This was a font class. Restore it.)

(FONTCLASS (pop NAME)

NAME))

((AND NAME (NOT (ZEROP SIZE)))

(FONTCREATE NAME SIZE (COND

((AND (fetch (CHARLOOKS CLBOLD)

of LOOKS)

(fetch (CHARLOOKS CLITAL)

of LOOKS))

'BOLDITALIC)

((fetch (CHARLOOKS CLBOLD)

of LOOKS)

'BOLD)

((fetch (CHARLOOKS CLITAL)

of LOOKS)

'ITALIC])

(\TEDIT.GET.PARALOOKS1

[LAMBDA (FILE)

; Edited 30-May-91 20:34 by jds

(* Read a paragraph format spec from the FILE, and return it for

later use.)

(PROG ((LOOKS (create FMTSPEC))

TABFLG DEFAULTTAB TABCOUNT TABS TABSPEC)

(replace (FMTSPEC 1STLEFTMAR) of LOOKS with (\SMALLPIN FILE))

(* Left margin for the first line of the paragraph)

(replace (FMTSPEC LEFTMAR) of LOOKS with (\SMALLPIN FILE))

(* Left margin for the rest of the paragraph)

(replace (FMTSPEC RIGHTMAR) of LOOKS with (\SMALLPIN FILE))

(* Right margin for the paragraph)

(replace (FMTSPEC LEADBEFORE) of LOOKS with (\SMALLPIN FILE))

(* Leading before the paragraph)

(replace (FMTSPEC LEADAFTER) of LOOKS with (\SMALLPIN FILE))

(* Lead after the paragraph)

(replace (FMTSPEC LINELEAD) of LOOKS with (\SMALLPIN FILE))

(* inter-line leading)

(replace (FMTSPEC TABSPEC) of LOOKS with (SETQ TABSPEC (CONS NIL NIL)))

(* Will be tab specs)

(SETQ TABFLG (\BIN FILE))

(replace (FMTSPEC QUAD) of LOOKS with (SELECTC (\BIN FILE)

(1 'LEFT)

(2 'RIGHT)

(3 'CENTERED)

(4 'JUSTIFIED)

(SHOULDNT)))

(COND

((NOT (ZEROP (LOGAND TABFLG 1)))

(* There are tabs to read)

(SETQ DEFAULTTAB (\SMALLPIN FILE))

(SETQ TABCOUNT (\BIN FILE))


```

[SETQ TABS (for TAB# from 1 to TABCOUNT collect (create TAB
                                                    TABX _ (\SMALLPIN FILE)
                                                    TABKIND _ (SELECTQ (\BIN FILE)
                                                                (0 'LEFT)
                                                                (1 'RIGHT)
                                                                (2 'CENTERED)
                                                                (3 'DECIMAL)
                                                                (SHOULDNT])

(OR (ZEROP DEFAULTTAB)
    (RPLACA TABSPEC DEFAULTTAB))
(RPLACD TABSPEC TABS)))
[COND
  ((NOT (ZEROP (LOGAND TABFLG 2))) (* There are other paragraph parameters to be read.)
    (replace (FMTSPEC FMTSPECIALX) of LOOKS with (\SMALLPIN FILE)) (* Special X location on page for this paragraph)
    (replace (FMTSPEC FMTSPECIALY) of LOOKS with (\SMALLPIN FILE))
    (replace (FMTSPEC FMTUSERINFO) of LOOKS with (\ARBIN FILE))
    (replace (FMTSPEC FMTPARATYPE) of LOOKS with (\ATMIN FILE))
    (replace (FMTSPEC FMTPARASUBTYPE) of LOOKS with (\ATMIN FILE))
    (replace (FMTSPEC FMTSTYLE) of LOOKS with (\ARBIN FILE))
    (replace (FMTSPEC FMTCHARSTYLES) of LOOKS with (\ARBIN FILE))
    (replace (FMTSPEC FMTNEWPAGEBEFORE) of LOOKS with (\ARBIN FILE))
    (replace (FMTSPEC FMTNEWPAGEAFTER) of LOOKS with (\ARBIN FILE))
  (RETURN LOOKS])

```

(TEDIT.GET.OBJECT1

[LAMBDA (STREAM PIECE FILE CURCH#)

; Edited 12-Jun-90 18:17 by mitani
(* Get an object from the file)

(* CURCH# = fileptr within the text section of the file where the object's text starts.)

```

(PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
      FILEPTRSAVE NAMELEN GETFN OBJ)
  (SETQ GETFN (\ATMIN FILE)) (* The GETFN for this kind of IMAGEOBJ)
  (SETQ FILEPTRSAVE (GETFILEPTR FILE)) (* Save our file location thru the building of the object)
  (SETFILEPTR FILE CURCH#)
  (SETQ OBJ (READIMAGEOBJ FILE GETFN))
  (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])
)

```

;; VERSION 0 Compatibility reading functions

(DEFINEQ

(TEDIT.BUILD.PCTB0

[LAMBDA (TEXT TEXTOBJ PCCOUNT START END)

; Edited 2-May-95 14:11 by sybalsky:mv:envos

;;; READ OBSOLETE FORMATS OF TEDIT FILE

```

(PROG [SEL LINES PCTB PC OLDPC TYPECODE PCLEN CHLOOKSSEEN NEWPC PARALOOKSSEEN PIECEINFOCH# CACHE TTEXTOBJ
      USER.CMFILE TSTREAM USERFILEFORMAT USERTEMP (CURFILECH# (OR START 0))
      (CURCH# 1)
      (TEXTSTREAM (AND TEXTOBJ (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
                                     ; Get the number of pieces needed (if AN formatted file),
                                     ; otherwise PCCOUNT will be NIL
      [SETQ DEFAULTPARALOOKS (OR DEFAULTPARALOOKS (COND
        (TEXTOBJ (fetch (TEXTOBJ FMTSPEC) of TEXTOBJ))
        (T (create FMTSPEC using TEDIT.DEFAULT.FMTSPEC))
      (SETQ PCTB (\MAKEPCTB NIL PCCOUNT))
      (replace (TEXTOBJ PCTB) of TEXTOBJ with PCTB)
      (SETFILEPTR TEXT (IDIFFERENCE (OR END (GETEOFPTR TEXT))
                                     8))
      (SETQ PIECEINFOCH# (\DWIN TEXT))
      (SETFILEPTR TEXT PIECEINFOCH#)
      (bind (OLDPC _ NIL) for I from 1 to PCCOUNT as PCN from \FirstPieceOffset by \EltsPerPiece
        do (SETQ PC (create PIECE
          PFILE _ TEXT
          PFPOS _ CURFILECH#
          PLEN _ (SETQ PCLEN (\DWIN TEXT))
          PREVPiece _ OLDPC
          PPARALOOKS _ DEFAULTPARALOOKS))
      [COND
        (OLDPC (replace (PIECE NEXTPIECE) of OLDPC with PC)
          (replace (PIECE PPARALOOKS) of PC with (fetch (PIECE PPARALOOKS) of OLDPC]

```

```

      (SETQ TYPECODE (\SMALLPIN TEXT))
      (SELECTC TYPECODE
        (\PieceDescriptorLOOKS
          (TEDIT.GET.CHARLOOKS0 PC TEXT)
          (add CURFILECH# (fetch (PIECE PLEN) of PC)))
        (\PieceDescriptorOBJECT
          (TEDIT.GET.OBJECT0 TEXTSTREAM PC TEXT CURFILECH#)
          (add CURFILECH# (fetch (PIECE PLEN) of PC))
          (replace (PIECE PLEN) of PC with 1) ; Only object--can't be followed by either of the others.
        )
        (\PieceDescriptorPARA
          (AND OLDPC (replace (PIECE PPARALAST) of OLDPC with T))
          (TEDIT.GET.PARALOOKS0 PC TEXT)
          (replace (PIECE PLEN) of PC with (\DWIN TEXT))
          (\SMALLPIN TEXT) ; Set this piece's length from the character looks.
          (TEDIT.GET.CHARLOOKS0 PC TEXT) ; Skip the piece-type code, since we know what's next
          (add CURFILECH# (fetch (PIECE PLEN) of PC)) ; This document is 'formatted' .
          (AND TEXTOBJ (replace (TEXTOBJ FORMATTEDP) of TEXTOBJ with T)))
        (SHOULDNT "Impossible piece-type code in BUILD.PCTB"))
      (SETQ OLDPC PC)
      (\INSERTPIECE PC 'LASTPIECE TEXTOBJ) ; * INSERT-BRT (CREATEPCNODE CURCH# PC) PCTB)
      (add CURCH# (fetch (PIECE PLEN) of PC))
    finally ; * INSERT-BRT (CREATEPCNODE CURCH#
      (QUOTE LASTPIECE)) PCTB))
  (RETURN PCTB))

```

(TEDIT.GET.CHARLOOKS0

[LAMBDA (PC FILE)

; Edited 30-May-91 20:26 by jds
 ; * Put a description of LOOKS into FILE.
 LOOKS apply to characters CH1 thru CHLM-1)

```

  (PROG (FONT STR NAME NAMELEN SIZE SUPER SUB PROPS STYLESTR USERSTR (LOOKS (create CHARLOOKS)))
    (replace (PIECE PLOOKS) of PC with LOOKS)
    (SETQ NAMELEN (\SMALLPIN FILE)) ; * The length of the description which follows
    [SETQ NAME (PACK (for I from 1 to NAMELEN collect (CHARACTER (\BIN FILE)
      ; * The font name
      ; * Size of the type, in points
      ; * Superscripting distance
      ; * former Subscripting distance
      (SETQ SIZE (\SMALLPIN FILE))
      (SETQ SUPER (\SMALLPIN FILE))
      (SETQ SUB (\SMALLPIN FILE))
      (OR (ZEROP SUB)
        (SETQ SUPER (IMINUS SUB)))
    ]
  )

```

(* If this is an old file, it'll have a subscript value not zero. Let those past and do the right thing.)

```

  (COND
    ((NOT (ZEROP (\BIN FILE))) ; * This text is NEW. Mark it so.)
    (replace (PIECE PNEW) of PC with T)))
  [COND
    ((NOT (ZEROP (\BIN FILE))) ; * There is style or user information to be read
      (SETQ STYLESTR (\STRINGIN FILE))
      (SETQ USERSTR (\STRINGIN FILE))
      (COND
        ((NOT (ZEROP (NCHARS STYLESTR))) ; * There IS style info
          (replace (CHARLOOKS CLSTYLE) of LOOKS with (READ STYLESTR))
          (T (replace (CHARLOOKS CLSTYLE) of LOOKS with 0)))
        (COND
          ((NOT (ZEROP (NCHARS USERSTR))) ; * There IS user info
            (replace (CHARLOOKS CLUSERINFO) of LOOKS with (READ USERSTR)
          )
        )
      )
    (SETQ PROPS (\SMALLPIN FILE))
    (with CHARLOOKS LOOKS [SETQ CLBOLD (NOT (ZEROP (LOGAND 512 PROPS)
      [SETQ CLITAL (NOT (ZEROP (LOGAND 256 PROPS)
      [SETQ CLULINE (NOT (ZEROP (LOGAND 128 PROPS)
      [SETQ CLOLINE (NOT (ZEROP (LOGAND 64 PROPS)
      [SETQ CLSTRIKE (NOT (ZEROP (LOGAND 32 PROPS)
      [SETQ CLSMALLCAP (NOT (ZEROP (LOGAND 16 PROPS)
      [SETQ CLPROTECTED (NOT (ZEROP (LOGAND 8 PROPS)
      [SETQ CLINVISIBLE (NOT (ZEROP (LOGAND 4 PROPS)
      [SETQ CLSELHERE (NOT (ZEROP (LOGAND 2 PROPS)
      [SETQ CLCANCOPY (NOT (ZEROP (LOGAND 1 PROPS)
      (SETQ CLSIZE SIZE)
      (SETQ CLOFFSET SUPER))
    (replace (CHARLOOKS CLFONT) of LOOKS with (AND NAME (NOT (ZEROP SIZE))
      (FONTCREATE NAME SIZE (COND
        ((AND (fetch (CHARLOOKS CLBOLD)
          of LOOKS)
          (fetch (CHARLOOKS CLITAL)
            of LOOKS))
        'BOLDITALIC)
        ((fetch (CHARLOOKS CLBOLD)
          of LOOKS)
        'BOLD)
        ((fetch (CHARLOOKS CLITAL)
          of LOOKS)
        'ITALIC])
      )
    ]
  )

```

(TEDIT.GET.OBJECTO

[LAMBDA (STREAM PIECE FILE CURCH#)

; Edited 12-Jun-90 18:17 by mitani
(* Get an object from the file)

(* CURCH# = fileptr within the text section of the file where the object's text starts.)

```

(PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
      FILEPTRSAVE NAMELEN GETFN OBJ)
  (SETQ GETFN (\ATMIN FILE))
  (SETQ FILEPTRSAVE (GETFILEPTR FILE))
  (SETFILEPTR FILE CURCH#)
  (SETQ OBJ (READIMAGEOBJ FILE GETFN))
  (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.UNIQIFY.CHARLOOKS (CHARLOOKS.FROM.FONT
                                       DEFAULTFONT)
                                       TEXTOBJ)]

  (RETURN (fetch (PIECE POBJ) of PIECE])

```

(* The GETFN for this kind of IMAGEOBJ)
(* Save our file location thru the building of the object)**(TEDIT.GET.PARALOOKSO**

[LAMBDA (PC FILE)

; Edited 30-May-91 20:34 by jds
(* Put a description of LOOKS into FILE.
LOOKS apply to characters CH1 thru CHLIM-1)

```

(PROG ((LOOKS (create FMTSPEC))
      TABFLG DEFAULTTAB TABCOUNT TABS TABSPEC)
  (replace (PIECE PPARALOOKS) of PC with LOOKS)
  (replace (FMTSPEC 1STLEFTMAR) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC LEFTMAR) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC RIGHTMAR) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC LEADBEFORE) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC LEADAFTER) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC LINELEAD) of LOOKS with (\SMALLPIN FILE))
  (replace (FMTSPEC TABSPEC) of LOOKS with (SETQ TABSPEC (CONS NIL NIL)))
  (SETQ TABFLG (\BIN FILE))
  (replace (FMTSPEC QUAD) of LOOKS with (SELECTC (\BIN FILE)
    (1 'LEFT)
    (2 'RIGHT)
    (3 'CENTERED)
    (4 'JUSTIFIED)
    (SHOULDNT)))
  (COND
    ((NOT (ZEROP TABFLG))
     (SETQ DEFAULTTAB (\SMALLPIN FILE))
     (SETQ TABCOUNT (\BIN FILE))
     [SETQ TABS (for TAB# from 1 to TABCOUNT collect (create TAB
       TABX _ (\SMALLPIN FILE)
       TABKIND _ (SELECTQ (\BIN FILE)
        (0 'LEFT)
        (1 'RIGHT)
        (2 'CENTERED)
        (3 'DECIMAL)
        (SHOULDNT)]
       (OR (ZEROP DEFAULTTAB)
           (RPLACA TABSPEC DEFAULTTAB))
       (RPLACD TABSPEC TABS])
    )

```

```

(PUTPROPS TEDITFILE COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992
                                1993 1994 1995))

```

FUNCTION INDEX

| | | | | | |
|---------------------------------|----|-------------------------------------|----|------------------------------------|----|
| TEDIT.BUILD.PCTB | 1 | TEDIT.RAW.INCLUDE | 10 | \TEDIT.GET.PARALOOKS1 | 32 |
| TEDIT.BUILD.PCTB0 | 33 | \ARBIN | 8 | \TEDIT.GET.SINGLE.CHARLOOKS | 18 |
| TEDIT.BUILD.PCTB1 | 30 | \ARBOUT | 17 | \TEDIT.GET.SINGLE.CHARLOOKS2 | 24 |
| TEDIT.BUILD.PCTB2 | 22 | \ATMIN | 8 | \TEDIT.GET.SINGLE.PARALOOKS | 20 |
| TEDIT.FORMATTEDFILEP | 6 | \ATMOUT | 17 | \TEDIT.GET.SINGLE.PARALOOKS2 | 26 |
| TEDIT.GET | 6 | \DWIN | 8 | \TEDIT.PUT.CHARLOOKS.LIST | 18 |
| TEDIT.GET.CHARLOOKS0 | 34 | \DWOUT | 17 | \TEDIT.PUT.CHARLOOKS.LIST2 | 30 |
| TEDIT.GET.OBJECT0 | 35 | \STRINGIN | 8 | \TEDIT.PUT.PARALOOKS.LIST | 21 |
| TEDIT.GET.OBJECT1 | 33 | \STRINGOUT | 17 | \TEDIT.PUT.PARALOOKS.LIST2 | 30 |
| TEDIT.GET.PAGEFRAMES1 | 31 | \TEDIT-OPEN-FONT-FILE | 17 | \TEDIT.PUT.SINGLE.CHARLOOKS | 19 |
| TEDIT.GET.PARALOOKS0 | 35 | \TEDIT.CONVERT.FOREIGN.FORMAT | 6 | \TEDIT.PUT.SINGLE.CHARLOOKS2 | 25 |
| TEDIT.INCLUDE | 9 | \TEDIT.FORMATTEDP1 | 8 | \TEDIT.PUT.SINGLE.PARALOOKS | 21 |
| TEDIT.PARSE.PAGEFRAMES1 | 8 | \TEDIT.GET.CHARLOOKS.LIST | 18 | \TEDIT.PUT.SINGLE.PARALOOKS2 | 24 |
| TEDIT.PUT | 12 | \TEDIT.GET.CHARLOOKS.LIST2 | 24 | \TEDIT.PUTRESET | 17 |
| TEDIT.PUT.PCTB | 13 | \TEDIT.GET.CHARLOOKS1 | 32 | \TEDIT.SET.WINDOW | 9 |
| TEDIT.PUT.PCTB2 | 26 | \TEDIT.GET.PARALOOKS.LIST | 20 | | |
| TEDIT.PUT.PIECE.DESRIPTOR | 17 | \TEDIT.GET.PARALOOKS.LIST2 | 26 | | |

VARIABLE INDEX

| | | | |
|------------------------------|----|---------------------------|----|
| *TEDIT-FILE-READTABLE* | 22 | TEDIT.INPUT.FORMATS | 22 |
|------------------------------|----|---------------------------|----|

CONSTANT INDEX

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