File created: 21-Feb-2025 09:49:05 **{WMEDLEY}<library>TEDIT>TEDIT-LOOKS.;392**

edit by: rmk

changes to: (FNS TEDIT.CARETLOOKS)

previous date: 19-Feb-2025 12:00:37 {WMEDLEY}<library>TEDIT>TEDIT-LOOKS.;390

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-LOOKSCOMS
       [;; Support for Character looks (font, italic/bold, sub/superscripting, etc) and paragraph looks (margins, centered/justified, tabs, etc.). Uses
        ;; compiled create functions in case DWIM is not available at loadup time.
        (DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS CHARLOOKS PARALOOKS)
                                                 (MACROS \WORDSETA)
                                                 (MACROS ONOFF)
                                                 (MACROS GETCLOOKS SETCLOOKS FGETCLOOKS FSETCLOOKS CHARLOOKS!)
                                                 (MACROS GETPLOOKS SETPLOOKS FGETPLOOKS FSETPLOOKS PARALOOKS!)
                                                                    ; TO BE REMOVED
                                                 (MACROS FSETPARA FGETPARA GETPARA SETPARA)))
        (INITRECORDS CHARLOOKS PARALOOKS PENDINGTAB)
        (FNS \TEDIT.CHARLOOKS.DEFPRINT \TEDIT.PARALOOKS.DEFPRINT)
        (COMS ;; Added by yabu.fx, for SUNLOADUP without DWIM.  Not sure any of these are needed/used.
              (FNS \TEDIT.CREATE.DEFAULT.FMTSPEC \TEDIT.CREATE.FACE.MENU \TEDIT.CREATE.SIZE.MENU))
        [INITVARS (TEDIT.DEFAULT.FOLIO)
               (TEDIT.KNOWN.FONTS '((Classic 'CLASSIC)
                                    (Modern 'MODERN)
                                    (Terminal 'TERMINAL)
                                    (Titan 'TITAN)
                                    (Gacha 'GACHA)
                                    (Helvetica 'HELVETICA)
                                    (Times% Roman 'TIMESROMAN]
        (VARS TEDIT.CHARLOOKS.FEATURES (TEDIT.DEFAULT.FMTSPEC (\TEDIT.CREATE.DEFAULT.FMTSPEC))
              (TEDIT.FACE.MENU (\TEDIT.CREATE.FACE.MENU))
              (TEDIT.SIZE.MENU (\TEDIT.CREATE.SIZE.MENU)))
        (FNS \TEDIT.CHARLOOK.FEATUREP)
        (GLOBALVARS TEDIT.CHARLOOKS.FEATURES TEDIT.KNOWN.FONTS TEDIT.FACE.MENU TEDIT.SIZE.MENU
               TEDIT.DEFAULT.FMTSPEC)
        (ADDVARS (FONTVARS (TEDIT.PROMPT.FONT DEFAULTFONT)
                           (TEDIT.ICON.FONT MENUFONT)))
        (COMS                                                   ; Character looks functions
              (FNS \TEDIT.CHARLOOKS.FROM.FONT \TEDIT.EQCLOOKS \TEDIT.SAMECLOOKS TEDIT.CARETLOOKS
                   TEDIT.COPY.LOOKS \TEDIT.UNPARSE.CHARLOOKS.LIST \TEDIT.MODIFYLOOKS TEDIT.NEW.FONT
                   \TEDIT.CARETLOOKS.VERIFY \TEDIT.CARETPIECE \TEDIT.GET.INSERT.CHARLOOKS
                   \TEDIT.GET.TERMSA.WIDTHS \TEDIT.PARSE.CHARLOOKS.LIST)
              (COMS (FNS \TEDIT.TRANSLATE.ASCIICHARS \TEDIT.CONVERT.TO.FORMATTED)
                    (MACROS \TEDIT.TRANSLATE.ASCII.CHARLOOKS))
              (FNS \TEDIT.UNIQUIFY.CHARLOOKS \TEDIT.UNIQUIFY.PARALOOKS \TEDIT.UNIQUIFY.ALL
                   \TEDIT.FLUSH.UNUSED.LOOKS)
              ;; Public entries
              (FNS TEDIT.LOOKS TEDIT.GET.LOOKS TEDIT.SUBLOOKS TEDIT.FINDLOOKS)
              [INITVARS (TEDIT.FONTCLASSES '(DISPLAY PDF POSTSCRIPT INTERPRESS PRESS]
              (FNS \TEDIT.CHANGE.CHARLOOKS \TEDIT.CHANGE.CHARLOOKS.NEW \TEDIT.CHARLOOKS.CHANGE.FONT \TEDIT.LOOKS
                   \TEDIT.FONTCOPY \TEDIT.COERCE.FONTCLASS))
        (COMS                                                   ; Paragraph looks functions
              (FNS \TEDIT.EQFMTSPEC TEDIT.GET.PARALOOKS \TEDIT.PARSE.PARALOOKS.LIST TEDIT.PARALOOKS
                   \TEDIT.CHANGE.PARALOOKS \TEDIT.CHANGE.PARALOOKS.NEW TEDIT.COPY.PARALOOKS \TEDIT.PARABOUNDS)
              ;; For making paragraph-looks substitutions.
              (FNS TEDIT.SUBPARALOOKS SAMEPARALOOKS))
        (FNS \TEDIT.MARK.REVISION)
                                                                ; Revision-mark support
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                            (NLAML)
                                                                            (LAMA])

;; Support for Character looks (font, italic/bold, sub/superscripting, etc) and paragraph looks (margins, centered/justified, tabs, etc.). Uses compiled
;; create functions in case DWIM is not available at loadup time.

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

[DATATYPE CHARLOOKS (;; Describes the appearance ("Looks") of characters in a TEdit document.
                     ;; NOTE:  If fields change EQCLOOKS should change too.
                     CLFONT                                     ; The font descriptor for these characters
```

```
                    CLFONTUNPARSE

                    ;; Name of the font (e.g., HELVETICA) THIS FIELD IS A HINT, OR FOR USE IN CHARLOOKS-BUILDING CODE.  USE
                    ;; FONTPROP TO GET THE RIGHT VALUE FROM CLFONT.

                    NIL                                     ; Was  CLSIZE. Font size, in points
                    (NIL FLAG)                              ; Was CLITAL: T if the characters are italic, else NIL
                    (NIL FLAG)                              ; Was CLBoldT if the characters are bold, else NIL
                    (CLULINE FLAG)                          ; T if the characters are to be underscored, else NIL
                    (CLOLINE FLAG)                          ; T if the characters are to be overscored, else NIL
                    (CLSTRIKE FLAG)                         ; T if the characters are to be struck thru, else nil.
                    CLOFFSET                                ; A superscripting offset in points (?) else NIL (SUBSCRIPTING
                                                            ; IF NEGATIVE.)
                    (CLSMALLCAP FLAG)                       ; T if small caps, else NIL
                    (CLINVERTED FLAG)                       ; T if the characters are to be shown white-on-black
                    (CLPROTECTED FLAG)                      ; T if chars can't be selected, else NIL
                    (CLINVISIBLE FLAG)                      ; T if TEDIT is to ignore these chars;  else NIL
                    (CLSELAFTER FLAG)                       ; T if TEDIT can put selection after this char (for menu fields).

                    ;; Was CLSELHERE.

                    (CLCANCOPY FLAG)

                    ;; T if this text can be selected for copying, even tho protected (it will become unprotected after the copy;  for Dribble/TTY
                    ;; interface)

                    (CLUNBREAKABLE FLAG)                    ; Spaces are treated as nonbreaking spaces
                    CLSTYLE                                 ; The style to be used in marking these characters;  overridden
                                                            ; by the other fields
                    CLUSERINFO                              ; Any information that an outsider wants to include
                    CLLEADER                                ; For creating dotted and other kinds of leader
                    CLRULES

                    ;; For arbitrarily-places horizontal rules.  List of pairs, of (widthinpts  . offsetfrombaselineinpts).  Should be taken account of
                    ;; in ascent/descent calcs.

                    (CLMARK FLAG)

                    ;; Used for a mark-&-sweep of looks at PUT time -- T means this set of looks really IS in use in the document

                    (CLSELBEFORE FLAG)                      ; T if TEDIT can put selection before this char (for menu fields).

                    )
       CLOFFSET _ 0 (INIT (DEFPRINT 'CHARLOOKS (FUNCTION \TEDIT.CHARLOOKS.DEFPRINT)))
       (ACCESSFNS (CLNAME (fetch (CHARLOOKS CLFONTUNPARSE) of DATUM)
                          (replace (CHARLOOKS CLFONTUNPARSE) of DATUM with NEWVALUE]

(DATATYPE PARALOOKS (;; Describe the paragraph formatting for a paragraph in a TEdit document.
                    1STLEFTMAR                              ; Left margin of the first line of the paragraph
                    LEFTMAR                                 ; Left margin of the rest of the lines in the paragraph
                    RIGHTMAR                                ; Right margin for the paragraph
                    LEADBEFORE                              ; Leading above the paragraph's first line, in points
                    LEADAFTER                               ; Leading below the paragraph's bottom line, in points.  NOT
                                                            ; IMPLEMENTED.
                    LINELEAD                                ; Leading between lines, in points.  This space is added BELOW
                                                            ; each line in the para when TEDIT.LINELEADING.BELOW,
                                                            ; otherwise above, which is how it is documented.
                    FMTBASETOBASE                           ; The baseline-to-baseline spacing between lines in this
                                                            ; paragraph.  THIS  OVERRIDES THE LINE LEADING
                    NIL                                     ; Was TABSPEC: The list of tabs for this paragraph, including
                                                            ; CAR for a default tab width
                    QUAD                                    ; How the para is formatted: one of LEFT, RIGHT, CENTERED,
                                                            ; JUSTIFIED
                    FMTSTYLE                                ; The STYLE that controls this paragraph's appearance
                    FMTCHARSTYLES                           ; The characterstyles that control the appearance of characters
                                                            ; in this para (maybe?  may be part of the fmtstyle.)
                    FMTUSERINFO                             ; Space for a PLIST of user info
                    FMTSPECIALX                             ; A special horizontal location on the printed page for this para.
                    FMTSPECIALY                             ; A special vertical location on the page for this para
                    (FMTHEADINGKEEP FLAG)                   ; This para should be kept with the top line or so of the next
                                                            ; para..
                    FMTPARATYPE                             ; What kind of para this is: TEXT, PAGEHEADING, whatever
                    FMTPARASUBTYPE                          ; Sub type of the type, e.g., what KIND of page heading this is.
                    FMTNEWPAGEBEFORE                        ; Start a new box (if T) or back up the page formatting tree to
                                                            ; make a new box of the type named in the value -- by going the
                                                            ; least distance back up the tree, then back down until you find
                                                            ; that kind of box.
                    FMTNEWPAGEAFTER                         ; Similarly
                    FMTKEEP                                 ; For information about how this paragraph is to be kept with
                                                            ; other paragraphs.
                    FMTCOLUMN                               ; For setting up side-by-side paragraphs easily ala BravoX
                    FMTVERTRULES                            ; For Keeping track of vertical rules in force
                    (FMTMARK FLAG)                          ; Used to keep track of which PARALOOKSs are really being
                                                            ; used -- a mark & collect is done just before a PUT, so that only
                                                            ; 'real' PARALOOKSs make it into the file
                                                            ; Used for a mark&sweep of para looks at PUT time -- T means
                                                            ; this looks really IS in use in the document, so it makes sense to
                                                            ; save it on the file.
                    (FMTHARDCOPY FLAG)                      ; T if this paragraph is to be displayed in hardcopy-format.
                    FMTREVISED                              ; T (or perhaps a revision level or revision-mark spec??) if this
                                                            ; paragraph is to be marked as changed on output.
```

```
                    FMTHARDCOPYSCALE                        ; The units-per-point (DSPSCALE) of the hardcopy stream that is
                                                            ; simulated in hardcopy-display mode (FMTHARDCOPY=T
                    FMTDEFAULTTAB                           ; Default tab in points)
                    FMTTABS)                                ; List of tabs (in points)
      (INIT (DEFPRINT 'PARALOOKS (FUNCTION \TEDIT.PARALOOKS.DEFPRINT)))
      LEADBEFORE _ 0 LEADAFTER _ 0 LINELEAD _ 0)
)

(/DECLAREDATATYPE 'CHARLOOKS
      '(POINTER POINTER POINTER FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER
            POINTER POINTER POINTER FLAG FLAG)
      ;; ---field descriptor list elided by lister---
      '16)

(DEFPRINT 'CHARLOOKS (FUNCTION \TEDIT.CHARLOOKS.DEFPRINT))

(/DECLAREDATATYPE 'PARALOOKS
      '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
            POINTER FLAG POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG FLAG POINTER POINTER
            POINTER POINTER)
      ;; ---field descriptor list elided by lister---
      '50)

(DEFPRINT 'PARALOOKS (FUNCTION \TEDIT.PARALOOKS.DEFPRINT))

(DECLARE%: EVAL@COMPILE

(PUTPROPS \WORDSETA DMACRO (OPENLAMBDA (A J V)
                              [CHECK (AND (ARRAYP A)
                                          (ZEROP (fetch (ARRAYP ORIG) of A))
                                          (EQ \ST.POS16 (fetch (ARRAYP TYP) of A]
                              (CHECK (IGREATERP (fetch (ARRAYP LENGTH) of A)
                                                J))
                              (\PUTBASE (fetch (ARRAYP BASE) of A)
                                     (IPLUS (fetch (ARRAYP OFFST) of A)
                                            J)
                                     V)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS ONOFF MACRO [OPENLAMBDA (VAL)
                        (COND
                           (VAL 'ON)
                           (T 'OFF])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETCLOOKS MACRO ((CL FIELD)
                           (fetch (CHARLOOKS FIELD) of CL)))

(PUTPROPS SETCLOOKS MACRO ((CL FIELD NEWVALUE)
                           (replace (CHARLOOKS FIELD) of CL with NEWVALUE)))

(PUTPROPS FGETCLOOKS MACRO ((CL FIELD)
                            (ffetch (CHARLOOKS FIELD) of CL)))

(PUTPROPS FSETCLOOKS MACRO ((CL FIELD NEWVALUE)
                            (freplace (CHARLOOKS FIELD) of CL with NEWVALUE)))

(PUTPROPS CHARLOOKS! MACRO ((CL)
                            (\DTEST CL 'CHARLOOKS)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETPLOOKS MACRO ((PLOOKS FIELD)
                           (fetch (PARALOOKS FIELD) of PLOOKS)))

(PUTPROPS SETPLOOKS MACRO ((PLOOKS FIELD NEWVALUE)
                           (replace (PARALOOKS FIELD) of PLOOKS with NEWVALUE)))

(PUTPROPS FGETPLOOKS MACRO ((PLOOKS FIELD)
                            (ffetch (PARALOOKS FIELD) of PLOOKS)))

(PUTPROPS FSETPLOOKS MACRO ((PLOOKS FIELD NEWVALUE)
                            (freplace (PARALOOKS FIELD) of PLOOKS with NEWVALUE)))

(PUTPROPS PARALOOKS! MACRO ((PL)
                            (\DTEST PL 'PARALOOKS)))
)

(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS FSETPARA MACRO ((PLOOKS FIELD NEWVALUE)
                          (freplace (PARALOOKS FIELD) of PLOOKS with NEWVALUE)))

(PUTPROPS FGETPARA MACRO ((PLOOKS FIELD)
                          (ffetch (PARALOOKS FIELD) of PLOOKS)))

(PUTPROPS GETPARA MACRO ((PLOOKS FIELD)
                         (fetch (PARALOOKS FIELD) of PLOOKS)))

(PUTPROPS SETPARA MACRO ((PLOOKS FIELD NEWVALUE)
                         (replace (PARALOOKS FIELD) of PLOOKS with NEWVALUE)))
)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'CHARLOOKS
       '(POINTER POINTER POINTER FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER
              POINTER POINTER POINTER FLAG FLAG)
```

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

```
       '16)

(DEFPRINT 'CHARLOOKS (FUNCTION \TEDIT.CHARLOOKS.DEFPRINT))

(/DECLAREDATATYPE 'PARALOOKS
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER FLAG POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG FLAG POINTER POINTER
              POINTER POINTER)
```

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

```
       '50)

(DEFPRINT 'PARALOOKS (FUNCTION \TEDIT.PARALOOKS.DEFPRINT))

(/DECLAREDATATYPE 'PENDINGTAB '(POINTER POINTER POINTER POINTER FULLXPOINTER POINTER)
```

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

```
       '12)

(DEFINEQ
```

(\TEDIT.CHARLOOKS.DEFPRINT

```
  [LAMBDA (LOOKS STREAM CPL NOLOC)                          ; Edited  2-Jan-2025 11:46 by rmk
                                                            ; Edited 26-Aug-2023 11:10 by rmk
```

;; CPL seems to be a hidden argument passed on calls from \PRINT-USING-DEFPRINT, usually with value 0. So NOLOC is one beyond that

```
    (LET* ((LOC (LOC LOOKS))
           (FONT (GETCLOOKS LOOKS CLFONT))
           (FACE (FONTPROP FONT 'FACE))
           INFO)
          [SETQ FACE (CONCATCODES (LIST (CHCON1 (CAR FACE))
                                        (CHCON1 (CADR FACE]
          (SETQ INFO (CONCAT (L-CASE (FONTPROP FONT 'FAMILY)
                                     T)
                             (FONTPROP FONT 'SIZE)
                             (CL:IF (STREQUAL FACE "MR")
                                 ""
                                 FACE)))
          (CONS (CL:IF NOLOC
                       INFO
                       (CONCAT "{CL" (CAR LOC)
                               "/"
                               (CDR LOC)
                               ":" INFO "}"))])
```

(\TEDIT.PARALOOKS.DEFPRINT

```
  [LAMBDA (PARALOOKS STREAM)                                ; Edited 19-Feb-2025 11:52 by rmk
                                                            ; Edited  8-Feb-2025 23:27 by rmk
                                                            ; Edited 26-Aug-2023 11:11 by rmk
    (LET ((LOC (LOC PARALOOKS)))
         (CONS (CONCAT "{PL" (CAR LOC)
                       "/"
                       (CDR LOC)
                       ":"
                       (SUBSTRING (GETPLOOKS PARALOOKS QUAD)
                              1 2)
                       "-"
                       (GETPLOOKS PARALOOKS LEFTMAR)
                       "-"
                       (GETPLOOKS PARALOOKS RIGHTMAR)
                       "}"])

)
```

;; Added by yabu.fx, for SUNLOADUP without DWIM. Not sure any of these are needed/used.

```
(DEFINEQ
```

(\TEDIT.CREATE.DEFAULT.FMTSPEC

```
  [LAMBDA NIL                                                    ; Edited  8-Feb-2025 22:05 by rmk
                                                                 ; Edited  4-Aug-2024 17:13 by rmk
                                                                 ; Edited 28-Jul-2024 12:57 by rmk
                                                                 ; Edited 24-Aug-2023 23:31 by rmk

    (create PARALOOKS
            QUAD _ 'LEFT
            1STLEFTMAR _ 0
            LEFTMAR _ 0
            RIGHTMAR _ 0
            LEADBEFORE _ 0
            LEADAFTER _ 0
            LINELEAD _ 0
            FMTDEFAULTTAB _ DEFAULTTAB])
```

(\TEDIT.CREATE.FACE.MENU

```
  [LAMBDA NIL
    (create MENU
            ITEMS _ '(Bold Italic Bold% Italic Regular)
            CENTERFLG _ T
            TITLE _ "Face:"])
```

(\TEDIT.CREATE.SIZE.MENU

```
  [LAMBDA NIL
    (create MENU
            ITEMS _ '(6 7 8 9 10 11 12 14 18 24 30 36)
            CENTERFLG _ T
            MENUROWS _ 4
            TITLE _ "Type Size:"])

)

(RPAQ? TEDIT.DEFAULT.FOLIO )

(RPAQ? TEDIT.KNOWN.FONTS
       '((Classic 'CLASSIC)
         (Modern 'MODERN)
         (Terminal 'TERMINAL)
         (Titan 'TITAN)
         (Gacha 'GACHA)
         (Helvetica 'HELVETICA)
         (Times% Roman 'TIMESROMAN)))

(RPAQQ TEDIT.CHARLOOKS.FEATURES (DEVICE FAMILY SIZE FACE ITALIC WEIGHT SLOPE BOLD EXPANSION FONT INVERTED
                                        INVISIBLE OFFSET OFFSETINCREMENT OVERLINE PROTECTED SELECTPOINT SELAFTER
                                        SELBEFORE SIZEINCREMENT SMALLCAPS STRIKEOUT STYLE SUBSCRIPT SUPERSCRIPT
                                        UNBREAKABLE UNDERLINE USERINFO OFFSETTYPE))

(RPAQ TEDIT.DEFAULT.FMTSPEC (\TEDIT.CREATE.DEFAULT.FMTSPEC))

(RPAQ TEDIT.FACE.MENU (\TEDIT.CREATE.FACE.MENU))

(RPAQ TEDIT.SIZE.MENU (\TEDIT.CREATE.SIZE.MENU))

(DEFINEQ
```

(\TEDIT.CHARLOOK.FEATUREP

```
  [LAMBDA (P)                                                    ; Edited 27-Jul-2024 17:33 by rmk
    (MEMB P TEDIT.CHARLOOKS.FEATURES])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.CHARLOOKS.FEATURES TEDIT.KNOWN.FONTS TEDIT.FACE.MENU TEDIT.SIZE.MENU TEDIT.DEFAULT.FMTSPEC)
)

(ADDTOVAR FONTVARS (TEDIT.PROMPT.FONT DEFAULTFONT)
                   (TEDIT.ICON.FONT MENUFONT))
```

;; Character looks functions

```
(DEFINEQ
```

(\TEDIT.CHARLOOKS.FROM.FONT

```
  [LAMBDA (FONT)                                                 ; Edited  2-Jan-2025 10:21 by rmk
                                                                 ; Edited 31-Dec-2024 23:33 by rmk
                                                                 ; Edited 28-Dec-2024 12:28 by rmk
                                                                 ; Edited 21-Dec-2024 00:12 by rmk
                                                                 ; Edited 16-Dec-2024 13:14 by rmk
                                                                 ; Edited 10-Aug-2024 16:15 by rmk
```

; Edited 15-Oct-2023 18:56 by rmk
; Edited 25-Aug-2023 20:03 by rmk
; Edited 30-May-91 21:45 by jds

;; We fill the charlooks with the Display font attributes, since that's what this will be used for. Hardcopy will create its own version.

;; FONT may be a fontclass only on calls from TEDIT.LOOKS, not from the charlooks menu. This is because the menu specifies the family
;; independent of the face and size properties. The fontclass is installed and the other properties is installed in the looks and will be saved on the
;; file, even though only its display properties will show up in the menu. The user has no way, other than getting the looks and inspecting, to know
;; what will happen with other devices.

```
(CL:UNLESS (FONTP FONT)
    (SETQ FONT (if (AND (LITATOM FONT)
                        (type? FONTCLASS (GETATOMVAL FONT)))
                   then (GETATOMVAL FONT)
                 else (FONTCREATE FONT))))
(CL:WHEN (type? FONTCLASS FONT)
    (SETQ FONT (\TEDIT.COERCE.FONTCLASS FONT)))
(create CHARLOOKS
       CLFONT _ FONT
       CLNAME _ (FONTUNPARSE FONT])
```

(\TEDIT.EQCLOOKS

```
[LAMBDA (CLOOK1 CLOOK2)
```

; Edited 2-Jan-2025 21:01 by rmk
; Edited 18-Oct-2024 22:29 by rmk
; Edited 11-Aug-2024 20:41 by rmk
; Edited 31-Jul-2024 00:05 by rmk
; Edited 1-Dec-2023 19:27 by rmk
; Edited 9-Nov-2023 00:46 by rmk
; Edited 24-Jul-2023 17:18 by rmk
; Edited 1-Jun-93 11:49 by sybalsky:mv:envos

;; Given two sets of CHARLOOKS, are they effectively the same?

```
(OR (EQ CLOOK1 CLOOK2)
    (AND (OR (EQ (FGETCLOOKS CLOOK1 CLFONT)
                 (FGETCLOOKS CLOOK2 CLFONT))
             (EQUAL (FGETCLOOKS CLOOK1 CLNAME)
                    (FGETCLOOKS CLOOK2 CLNAME)))
         (EQ (FGETCLOOKS CLOOK1 CLPROTECTED)
             (FGETCLOOKS CLOOK2 CLPROTECTED))
         (EQ (FGETCLOOKS CLOOK1 CLINVISIBLE)
             (FGETCLOOKS CLOOK2 CLINVISIBLE))
         (EQ (FGETCLOOKS CLOOK1 CLSELAFTER)
             (FGETCLOOKS CLOOK2 CLSELAFTER))
         (EQ (FGETCLOOKS CLOOK1 CLSELBEFORE)
             (FGETCLOOKS CLOOK2 CLSELBEFORE))
         (EQ (FGETCLOOKS CLOOK1 CLCANCOPY)
             (FGETCLOOKS CLOOK2 CLCANCOPY))
         (EQ (FGETCLOOKS CLOOK1 CLULINE)
             (FGETCLOOKS CLOOK2 CLULINE))
         (EQ (FGETCLOOKS CLOOK1 CLOLINE)
             (FGETCLOOKS CLOOK2 CLOLINE))
         (EQ (FGETCLOOKS CLOOK1 CLINVERTED)
             (FGETCLOOKS CLOOK2 CLINVERTED))
         (EQ (FGETCLOOKS CLOOK1 CLSTRIKE)
             (FGETCLOOKS CLOOK2 CLSTRIKE))
         (EQ (FGETCLOOKS CLOOK1 CLOFFSET)
             (FGETCLOOKS CLOOK2 CLOFFSET))
         (EQ (FGETCLOOKS CLOOK1 CLSMALLCAP)
             (FGETCLOOKS CLOOK2 CLSMALLCAP))
         (EQUAL (FGETCLOOKS CLOOK1 CLSTYLE)
                (FGETCLOOKS CLOOK2 CLSTYLE))
         (EQ (FGETCLOOKS CLOOK1 CLUNBREAKABLE)
             (FGETCLOOKS CLOOK2 CLUNBREAKABLE))
         (EQUAL (FGETCLOOKS CLOOK1 CLUSERINFO)
                (FGETCLOOKS CLOOK2 CLUSERINFO])
```

(\TEDIT.SAMECLOOKS

```
[LAMBDA (CLOOK1 CLOOK2 FEATURES)
```

; Edited 2-Jan-2025 20:31 by rmk
; Edited 31-Dec-2024 23:59 by rmk
; Edited 31-Jul-2024 00:06 by rmk
; Edited 24-Jul-2023 17:17 by rmk
; Edited 30-May-91 21:45 by jds

;; Predicate to determine if CLOOK1 and CLOOK2 are the same in all the characteristics listed in FEATURES

```
(for F (FONT1 _ (FGETCLOOKS CLOOK1 CLFONT))
    (FONT2 _ (FGETCLOOKS CLOOK2 CLFONT)) in FEATURES always (SELECTQ F
                                                                (FAMILY (EQ (FONTPROP FONT1 'FAMILY)
                                                                            (FONTPROP FONT2 'FAMILY)))
                                                                (SIZE (EQ (FONTPROP FONT1 'SIZE)
                                                                          (FONTPROP FONT2 'SIZE)))
                                                                (EXPANSION (EQ (FONTPROP FONT1 'EXPANSION)
                                                                               (FONTPROP FONT2 'EXPANSION)))
                                                                (SLOPE (EQ (FONTPROP FONT1 'SLOPE)
                                                                           (FONTPROP FONT2 'SLOPE)))
                                                                (WEIGHT (EQ (FONTPROP FONT1 'WEIGHT)
                                                                            (FONTPROP FONT2 'WEIGHT)))
```

```
                                                    (SUPERSCRIPT (EQ (FGETCLOOKS CLOOK1 CLOFFSET
                                                                     )
                                                                     (FGETCLOOKS CLOOK2 CLOFFSET
                                                                     )))
                                                    (INVISIBLE (EQ (FGETCLOOKS CLOOK1
                                                                           CLINVISIBLE)
                                                                   (FGETCLOOKS CLOOK2
                                                                           CLINVISIBLE)))
                                                    (SELECTPOINT (EQ (FGETCLOOKS CLOOK1
                                                                           CLSELAFTER)
                                                                     (FGETCLOOKS CLOOK2
                                                                           CLSELAFTER)))
                                                    (PROTECTED (EQ (FGETCLOOKS CLOOK1
                                                                           CLPROTECTED)
                                                                   (FGETCLOOKS CLOOK2
                                                                           CLPROTECTED)))
                                                    (OVERLINE (EQ (FGETCLOOKS CLOOK1 CLOLINE)
                                                                  (FGETCLOOKS CLOOK2 CLOLINE)))
                                                    (STRIKEOUT (EQ (FGETCLOOKS CLOOK1 CLSTRIKE)
                                                                   (FGETCLOOKS CLOOK2 CLSTRIKE))
)
                                                    (UNDERLINE (EQ (FGETCLOOKS CLOOK1 CLULINE)
                                                                   (FGETCLOOKS CLOOK2 CLULINE)))
                                                    (UNBREAKABLE (FGETCLOOKS CLOOK1
                                                                         CLUNBREAKABLE)
                                                                 (FGETCLOOKS CLOOK2
                                                                         CLUNBREAKABLE))
                                                    (FACE (EQUAL (FONTPROP FONT1 'FACE)
                                                                 (FONTPROP FONT2 'FACE)))
                                                    (ERROR (CONCAT F " is an unknown feature of
                                                                   character looks.  Detected in
                                                                   SAMECLOOKS"])
```

(**TEDIT.CARETLOOKS**

```
  [LAMBDA (STREAM LOOKS)                                    ; Edited 21-Feb-2025 09:48 by rmk
                                                            ; Edited 15-Oct-2023 17:12 by rmk
                                                            ; Edited 28-May-2023 14:15 by rmk
                                                            ; Edited  6-Apr-2023 21:42 by rmk
                                                            ; Edited  8-Sep-2022 11:25 by rmk
                                                            ; Edited 30-May-91 21:40 by jds
    ;; Set the 'Caret looks' for a TEdit document, i.e., the looks that will be applied to newly-typed characters from here on.  Returns the previous caret
    ;; looks
    (LET ((TEXTOBJ (TEXTOBJ STREAM)))                       ; Parse up the looks he gave us, to make sure they're a valid
                                                            ; CHARLOOKS
         (PROG1 (FGETTOBJ TEXTOBJ CARETLOOKS)
                (change (FGETTOBJ TEXTOBJ CARETLOOKS)
                        (\TEDIT.CARETLOOKS.VERIFY TEXTOBJ (\TEDIT.PARSE.CHARLOOKS.LIST LOOKS DATUM TEXTOBJ))))])
```

(**TEDIT.COPY.LOOKS**

```
  [LAMBDA (STREAM SOURCE DEST)                              ; Edited 25-Nov-2024 14:38 by rmk
                                                            ; Edited  2-Aug-2024 08:47 by rmk
                                                            ; Edited 13-Jul-2024 23:15 by rmk
                                                            ; Edited 12-Jul-2024 00:37 by rmk
                                                            ; Edited 29-Apr-2024 13:00 by rmk
                                                            ; Edited 17-Mar-2024 00:27 by rmk
                                                            ; Edited  9-Feb-2024 11:42 by rmk
                                                            ; Edited 18-Apr-2023 23:53 by rmk
                                                            ; Edited 22-Oct-2022 15:27 by rmk
                                                            ; Edited 22-Aug-2022 13:14 by rmk
                                                            ; Edited 30-May-91 21:43 by jds
    ;; Copy the CHARACTER LOOKS of one piece of text (actually, the first selected character) to another piece of text.
    ;; According to (slightly wrong) documentation:
    ;; STREAM is the stream of the destination, no matter what.
    ;; STREAM is the stream of the source if SOURCE is an integer (or if it has no textstream). Otherwise, it provides its own stream
    ;; Not clear why the destination can't be in a different stream
    (SETQ STREAM (TEXTSTREAM STREAM))
    (LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of STREAM))
          SOURCESTREAM TOOBJ)                               ; get the character looks of the first character of SOURCE
         (if (type? SELECTION SOURCE)
             then (SETQ SOURCESTREAM (OR (GETSEL SOURCE SELTEXTSTREAM)
                                         STREAM))
           elseif (FIXP SOURCE)
             then (SETQ SOURCESTREAM STREAM)
                  (SETQ SOURCE (\TEDIT.UPDATE.SEL (\TEDIT.COPYSEL (TEXTSEL TEXTOBJ))
                                                  SOURCE 1))
           else (\ILLEGAL.ARG SOURCE))
         (if (type? SELECTION DEST)
             then                                           ; make sure that the destination selection is in this document;
                  (CL:UNLESS (OR (EQ STREAM (FGETSEL DEST SELTEXTSTREAM))
                                 (NULL (FGETSEL DEST SELTEXTSTREAM)))
                         (\LISPERROR "Destination selection is not in stream " STREAM))
```

```
          elseif (FIXP DEST)
            then (SETQ DEST (\TEDIT.UPDATE.SEL (\TEDIT.COPYSEL (TEXTSEL TEXTOBJ))
                                    DEST 1))
          else (\ILLEGAL.ARG DEST))
        (\TEDIT.CHANGE.CHARLOOKS STREAM (PCHARLOOKS (\TEDIT.CHTOPC (GETSEL SOURCE CH#)
                                                                   SOURCESTREAM))
              DEST])
```

(\TEDIT.UNPARSE.CHARLOOKS.LIST

```
[LAMBDA (LOOKS)                                                    ; Edited 29-Dec-2024 12:14 by rmk
                                                                   ; Edited 31-Jul-2024 00:06 by rmk
                                                                   ; Edited 24-Jul-2023 17:28 by rmk
                                                                   ; Edited 11-Feb-2023 14:51 by rmk
                                                                   ; Edited 30-May-91 21:45 by jds
```

;; Convert a CHARLOOKS into an equivalent PList-form for external consumption

```
(\DTEST LOOKS 'CHARLOOKS)
(LET (NEWLOOKS (OFFSET (FGETCLOOKS LOOKS CLOFFSET))
               (FONT (FGETCLOOKS LOOKS CLFONT)))
     [SETQ NEWLOOKS (NCONC (if (ILESSP OFFSET 0)
                               then (LIST 'SUBSCRIPT (IMINUS OFFSET))
                             else (IGREATERP OFFSET 0)
                               then (LIST 'SUPERSCRIPT OFFSET))
                           (for PVAL in (LIST (ONOFF (FGETCLOOKS LOOKS CLINVERTED))
                                              (ONOFF (FGETCLOOKS LOOKS CLULINE))
                                              (ONOFF (FGETCLOOKS LOOKS CLSTRIKE))
                                              (ONOFF (FGETCLOOKS LOOKS CLOLINE))
                                              (ONOFF (FGETCLOOKS LOOKS CLUNBREAKABLE))
                                              (ONOFF (FGETCLOOKS LOOKS CLPROTECTED))
                                              (ONOFF (FGETCLOOKS LOOKS CLSELAFTER))
                                              (ONOFF (FGETCLOOKS LOOKS CLINVISIBLE))
                                              (FGETCLOOKS LOOKS CLSTYLE)
                                              (FGETCLOOKS LOOKS CLUSERINFO LOOKS))
                              as PNAME
                              in '(INVERTED UNDERLINE STRIKEOUT OVERLINE UNBREAKABLE PROTECTED SELECTPOINT
                                           INVISIBLE STYLE USERINFO)
                              join (LIST PNAME PVAL]
```

;; Font properties. Don't show the separate properties if a font class, just the class. And if not a class, just show the properties, not the font.
;; So there is always a consistent picture.

```
     (SETQ NEWLOOKS (NCONC (if (type? FONTCLASS FONT)
                               then `(FONT ,FONT)
                             else (for PNAME in '(FAMILY SIZE WEIGHT SLOPE EXPANSION) as PVAL
                                     in (LIST (FONTPROP FONT 'FAMILY)
                                              (FONTPROP FONT 'SIZE)
                                              (FONTPROP FONT 'WEIGHT)
                                              (FONTPROP FONT 'SLOPE)
                                              (FONTPROP FONT 'EXPANSION))
                                     join (LIST PNAME PVAL)))
                           NEWLOOKS))
     NEWLOOKS])
```

(\TEDIT.MODIFYLOOKS

```
[LAMBDA (LINE STARTX DS LOOKS LINEBASEY)                           ; Edited 20-Nov-2023 14:18 by rmk
                                                                   ; Edited 27-May-2023 12:11 by rmk
                                                                   ; Edited 24-Sep-2022 11:12 by rmk
                                                                   ; Edited 30-May-91 21:45 by jds
```

;; Modify the screen to allow for underlining, etc. Also, restore the vertical offset to the baseline.

```
(LET ((CURX (DSPXPOSITION NIL DS))
      (CURY (DSPYPOSITION NIL DS))
      (FONT (fetch (CHARLOOKS CLFONT) of LOOKS)))
     (CL:WHEN (fetch (CHARLOOKS CLULINE) of LOOKS)             ; It's underlined.
         (MOVETO STARTX (ADD1 (IDIFFERENCE (IPLUS CURY)
                                           (GETLD LINE LTRUEDESCENT)))
                 DS)
         (RELDRAWTO (IDIFFERENCE CURX STARTX)
                 0 1 'PAINT DS))
     (CL:WHEN (fetch (CHARLOOKS CLOLINE) of LOOKS)             ; Over-line
         (MOVETO STARTX [IPLUS CURY (SUB1 (FONTPROP FONT 'ASCENT]
                 DS)
         (RELDRAWTO (IDIFFERENCE CURX STARTX)
                 0 1 'PAINT DS))
     (CL:WHEN (fetch (CHARLOOKS CLSTRIKE) of LOOKS)            ; Struck-thru
         (MOVETO STARTX (IPLUS CURY (IQUOTIENT (FONTPROP FONT 'ASCENT)
                                               3))
                 DS)
         (RELDRAWTO (IDIFFERENCE CURX STARTX)
                 0 1 'PAINT DS))
     (CL:WHEN (fetch (CHARLOOKS CLINVERTED) of LOOKS)          ; Inverse video
         (BLTSHADE BLACKSHADE DS STARTX (IDIFFERENCE CURY (FONTPROP FONT 'DESCENT))
                 (IDIFFERENCE CURX STARTX)
                 (FONTPROP FONT 'HEIGHT)
                 'INVERT))
     (MOVETO CURX LINEBASEY DS])
```

(TEDIT.NEW.FONT

```
[LAMBDA (TEXTOBJ)                                                    ; Edited 29-Jun-2024 16:31 by rmk
                                                                     (* jds " 8-Feb-85 11:27")
  (LET [(NAME (\TEDIT.MAKEFILENAME (TEDIT.GETINPUT TEXTOBJ "Name of font:  "]
       (CL:WHEN NAME
           [SETQ TEDIT.KNOWN.FONTS (NCONC1 TEDIT.KNOWN.FONTS (LIST NAME (KWOTE (U-CASE NAME]
           (U-CASE NAME))])
```

(\TEDIT.CARETLOOKS.VERIFY

```
[LAMBDA (TEXTOBJ NEWLOOKS)                                           ; Edited 15-Oct-2023 20:13 by rmk
                                                                     ; Edited 30-May-91 21:41 by jds

  ;; Check with the user's CARETLOOKSFN to see if he wants to make changes

  (LET ((CARETFN (GETTEXTPROP TEXTOBJ 'CARETLOOKSFN))
        LOOKS)
       (SETQ LOOKS (AND CARETFN (APPLY* CARETFN NEWLOOKS TEXTOBJ)))
       (if (EQ LOOKS 'DON'T)
           then                                                      ; He said not to change the looks.
                (OR (FGETTOBJ TEXTOBJ CARETLOOKS)
                    (FGETTOBJ TEXTOBJ DEFAULTCHARLOOKS))
         else (\TEDIT.UNIQUIFY.CHARLOOKS (OR LOOKS NEWLOOKS)
                     TEXTOBJ])
```

(\TEDIT.CARETPIECE

```
[LAMBDA (TEXTOBJ)                                                    ; Edited 17-Mar-2024 00:27 by rmk
                                                                     ; Edited  6-Apr-2023 21:32 by rmk

  (\TEDIT.CHTOPC (TEDIT.GETPOINT TEXTOBJ)
         TEXTOBJ])
```

(\TEDIT.GET.INSERT.CHARLOOKS

```
[LAMBDA (TEXTOBJ SEL/CHNO)                                           ; Edited 26-Nov-2024 04:58 by rmk
                                                                     ; Edited 23-Oct-2024 00:04 by rmk
                                                                     ; Edited 31-Jul-2024 12:10 by rmk
                                                                     ; Edited 17-Mar-2024 00:27 by rmk
                                                                     ; Edited 16-Feb-2024 22:48 by rmk
                                                                     ; Edited 15-Dec-2023 08:40 by rmk
                                                                     ; Edited  3-Aug-2023 22:39 by rmk
                                                                     ; Edited  9-Oct-2022 13:57 by rmk
                                                                     ; Edited 22-Aug-2022 13:21 by rmk
                                                                     ; Edited 30-May-91 21:45 by jds

  ;; We want to get the looks of a selected character. If point is RIGHT, that's the last character of the selection.  If LEFT, the first character of the
  ;; selection.

  ;; Return the looks at SEL, or defaults. Reset CLPROTECTED if need be.

  (LET ((PC (\TEDIT.CHTOPC (IMAX 1 (IMIN (TEXTLEN TEXTOBJ)
                                         (if (type? SELECTION SEL/CHNO)
                                             then (SELECTQ (GETSEL SEL/CHNO POINT)
                                                           (LEFT (ADD1 (TEDIT.GETPOINT TEXTOBJ SEL/CHNO)))
                                                           (RIGHT (SUB1 (TEDIT.GETPOINT TEXTOBJ SEL/CHNO)))
                                                           (\TEDIT.THELP "BAD POINT"))
                                           else SEL/CHNO)))
                    TEXTOBJ))
        LOOKS)
       (SETQ LOOKS (if PC
                       then (PCHARLOOKS PC)
                      elseif (FGETTOBJ TEXTOBJ DEFAULTCHARLOOKS)
                      else (\TEDIT.CHARLOOKS.FROM.FONT DEFAULTFONT)))
       (CL:WHEN (GETCLOOKS LOOKS CLPROTECTED)                        ; Unprotect by copying to a new CHARLOOKS.
           (SETQ LOOKS (create CHARLOOKS using LOOKS CLPROTECTED _ NIL CLSELAFTER _ NIL)))
       (\TEDIT.CARETLOOKS.VERIFY TEXTOBJ LOOKS])
```

(\TEDIT.GET.TERMSA.WIDTHS

```
[LAMBDA (TERMSA FONT)                                                (* jds "22-OCT-83 21:36")

        (* If the guy is using a terminal table, get an updated set of widths to reflect that.)

  (PROG ((NWIDTHS (ARRAY 256 'SMALLP 0 0)))
        (for I from 0 to 255 do (\WORDSETA NWIDTHS I (TEDIT.CHARWIDTH I FONT TERMSA)))
        (RETURN NWIDTHS])
```

(\TEDIT.PARSE.CHARLOOKS.LIST

```
[LAMBDA (NEWLOOKS DEFAULTCLOOKS TEXTOBJ)                             ; Edited 10-Aug-2024 23:52 by rmk
                                                                     ; Edited 31-Jul-2024 12:10 by rmk
                                                                     ; Edited 13-Nov-2023 01:08 by rmk
                                                                     ; Edited 11-Nov-2023 16:09 by rmk
                                                                     ; Edited 16-Oct-2023 09:02 by rmk
                                                                     ; Edited 24-Jul-2023 17:24 by rmk
                                                                     ; Edited 30-May-91 21:46 by jds

  ;; NEWLOOKS is either a CHARLOOKS, a FONTDESCRIPTOR, or a PLIST-format looks spec.  If NEWLOOKS is not already a CHARLOOKS, it
  ;; is coerced into one.
```

```
    (if (type? CHARLOOKS NEWLOOKS)
        then NEWLOOKS
      elseif (FONTP NEWLOOKS)
        then (\TEDIT.CHARLOOKS.FROM.FONT NEWLOOKS)
      else (\TEDIT.CHANGE.CHARLOOKS.NEW NEWLOOKS DEFAULTCLOOKS TEXTOBJ])

)

(DEFINEQ

(\TEDIT.TRANSLATE.ASCIICHARS
  [LAMBDA (TEXTOBJ NOASCIIFONTS)                                    ; Edited  2-Jan-2025 23:30 by rmk
                                                                    ; Edited 30-Dec-2024 21:30 by rmk
                                                                    ; Edited 22-Dec-2024 11:42 by rmk
                                                                    ; Edited 20-Dec-2024 13:34 by rmk
                                                                    ; Edited 23-Sep-2024 00:50 by rmk
                                                                    ; Edited 17-Mar-2024 00:25 by rmk
                                                                    ; Edited  1-Dec-2023 22:28 by rmk
                                                                    ; Edited 27-Nov-2023 16:13 by rmk
                                                                    ; Edited 26-Nov-2023 11:19 by rmk
                                                                    ; Edited 14-Nov-2023 19:21 by rmk
                                                                    ; Edited  9-Nov-2023 23:56 by rmk
    ;; Converts characters in Alto/Ascii font pieces to their XCCS character and font (more or less) equivalents.  The affected characters are put in their
    ;; own string pieces with their new CHARLOOKS.  Asciifont pieces are completely replaced if NOASCIIFONTS, otherwise untranslated characters
    ;; remain in their Asciifonts.

    ;; ASCIITONSTRANSLATIONS and the mapping arrays are from INTERPRESS.

    ;; \ASCII2MCCS is the default translation array, for Gacha, Timesroman.  HIPPO, MATH ... have their own.

    (DECLARE (GLOBALVARS ASCIITONSTRANSLATIONS \ASCII2MCCS))
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (CL:WHEN (thereis CL FAMILY in (FGETTOBJ TEXTOBJ TXTCHARLOOKSLIST)
                unless [EQ 'CLASSIC (SETQ FAMILY (FONTPROP (GETCLOOKS CL CLFONT)
                                                           'FAMILY]
                suchthat
                        ;; CLASSIC is in the list presumably to provide a coercion to MODERN for Interpress. We don't want to translate it.
                        (ASSOC FAMILY ASCIITONSTRANSLATIONS))
        (for CHNO CLOOKS TRANS MAPARRAY NEWFONTNAME STRING FAT CLOOKSLIST FAMILY TARRAYLAST from 1
           by (PLEN PC) as PC inpieces (\TEDIT.FIRSTPIECE TEXTOBJ) eachtime (SETQ CLOOKS (PLOOKS PC))
                                                                    (SETQ FAMILY (FONTPROP (GETCLOOKS
                                                                                                CLOOKS
                                                                                                CLFONT)
                                                                                           'FAMILY))
           unless (OR (EQ OBJECT.PTYPE (PTYPE PC))
                      (EQ FAMILY 'CLASSIC))
           when (SETQ TRANS (ASSOC FAMILY ASCIITONSTRANSLATIONS))
           do ;; PC needs some work.
              (SETQ MAPARRAY (CADR TRANS))
              (SETQ NEWFONTNAME (CADDR TRANS))
              (CL:WHEN MAPARRAY                                     ; Idiosyncratic fonts (MATH, CYRILLIC).
                  (SETQ MAPARRAY (GETATOMVAL MAPARRAY))             ; Global value
                  (CL:WHEN (AND NOASCIIFONTS (PREVPIECE PC))
                      ;; Look backward for NEWFONTNAME, since that piece has already been coerced.  The idea is to get Cyrillic to continue
                      ;; the previous looks (serif, san-serif)
                      (SETQ NEWFONTNAME (FONTPROP (GETCLOOKS (PLOOKS (PREVPIECE PC))
                                                             CLFONT)
                                                  'FAMILY))))
              (if (OR MAPARRAY NOASCIIFONTS)
                  then ;; Translate all characters in idiosyncratic fonts, flush everything and change the looks even for Helvetica etc. if NO
                       ;; ALTOFONTS
                       (CL:UNLESS MAPARRAY (SETQ MAPARRAY \ASCII2MCCS))
                       (SETQ TARRAYLAST (SUB1 (ARRAYSIZE MAPARRAY)))
                       ;; Create a string with the translated codes, then convert the existing piece to a string piece holding that string.
                       (SETQ STRING (ALLOCSTRING (PLEN PC)))
                       (for OFFSET OLDCODE NEWCODE from 1 to (PLEN PC)
                          do ;; Out-of-range alone and zero newcodes alone (some arrays are not filled in).
                             (SETQ OLDCODE (\TEDIT.PIECE.NTHCHARCODE TEXTOBJ PC OFFSET))
                             (RPLCHARCODE STRING OFFSET (if [OR (IGREATERP OLDCODE TARRAYLAST)
                                                                (ZEROP (SETQ NEWCODE (ELT MAPARRAY OLDCODE]
                                                            then OLDCODE
                                                            else NEWCODE)))
                       (SETQ FAT (ffetch (STRINGP FATSTRINGP) of STRING))
                       (FSETPC PC PTYPE (CL:IF FAT
                                               FATSTRING.PTYPE
                                               THINSTRING.PTYPE))
                       (FSETPC PC PCONTENTS STRING)
                       (FSETPC PC PFPOS NIL)
                       (FSETPC PC PBINABLE (NOT FAT))
                       (FSETPC PC PBYTESPERCHAR (CL:IF FAT
```

```
                                        1))
                (FSETPC PC PBYTELEN (CL:IF FAT
                                         (UNFOLD (PLEN PC)
                                                 2)
                                         (PLEN PC)))
                (FSETPC PC PLOOKS (\TEDIT.TRANSLATE.ASCII.CHARLOOKS TEXTOBJ CLOOKS NEWFONTNAME))
```

else ;; Must be a text font (GACHA, TIMESROMAN, HELVETICA) \ASCIITONS is the translation array, mostly identities.

;; Find the first change quickly, in piece coordinates. Then change whatever else needs it, slowly, in document coordinates.
;; It would be more complicated to do the replacements in piece coordinates, because the pieces would get split on the fly.

```
(for OFFSET OLDCODE NEWLOOKS from 1 to (PLEN PC) eachtime (SETQ OLDCODE (
                                                                \TEDIT.PIECE.NTHCHARCODE
                                                                  TEXTOBJ PC OFFSET))
   when (ILEQ OLDCODE 255) unless (EQ OLDCODE (ELT \ASCII2MCCS OLDCODE))
   do
```

;; First hit, scan/change the rest of PC

```
      (SETQ NEWLOOKS (\TEDIT.TRANSLATE.ASCII.CHARLOOKS TEXTOBJ CLOOKS NEWFONTNAME))
      (for I NEWCODE from (IPLUS CHNO (SUB1 OFFSET)) to (SUB1 (IPLUS CHNO (PLEN PC)))
         eachtime (SETQ OLDCODE (TEDIT.NTHCHARCODE TEXTOBJ I)) when (ILEQ OLDCODE 255)
         unless (EQ OLDCODE (SETQ NEWCODE (ELT \ASCII2MCCS OLDCODE)))
         do (TEDIT.RPLCHARCODE TEXTOBJ I NEWCODE NEWLOOKS))
      (RETURN)))
```

finally ;; Here we change the default and caret looks. Perhaps this should be done only if NOASCIIFONTS. But there is a risk that Ascii
;; fonts and characters would slip in by future editing.

```
(CL:WHEN NOASCIIFONTS
    (SETQ CLOOKS (FGETTOBJ TEXTOBJ DEFAULTCHARLOOKS))
    (SETQ FAMILY (FONTPROP (GETCLOOKS CLOOKS CLFONT)
                           'FAMILY))
    (CL:WHEN (AND (NEQ FAMILY 'CLASSIC)
                  (SETQ TRANS (ASSOC FAMILY ASCIITONSTRANSLATIONS)))
        (FSETTOBJ TEXTOBJ DEFAULTCHARLOOKS (\TEDIT.TRANSLATE.ASCII.CHARLOOKS TEXTOBJ CLOOKS
                                                    (CADDR TRANS))))
    (SETQ CLOOKS (FGETTOBJ TEXTOBJ CARETLOOKS))
    (SETQ FAMILY (FONTPROP (GETCLOOKS CLOOKS CLFONT)
                           'FAMILY))
    (CL:WHEN (AND (NEQ FAMILY 'CLASSIC)
                  (SETQ TRANS (ASSOC FAMILY ASCIITONSTRANSLATIONS)))
        (FSETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.TRANSLATE.ASCII.CHARLOOKS TEXTOBJ CLOOKS
                                                (CADDR TRANS)))))
(CL:WHEN CLOOKSLIST
```

;; Something happened, get rid of any lingering old looks

```
    (\TEDIT.UNIQUIFY.ALL TEXTOBJ))))])
```

(\TEDIT.CONVERT.TO.FORMATTED

```
[LAMBDA (TSTREAM START END)                                    ; Edited  7-Jul-2024 09:06 by rmk
                                                               ; Edited 10-May-2024 22:42 by rmk
                                                               ; Edited  6-May-2024 23:49 by rmk
                                                               ; Edited 29-Apr-2024 10:42 by rmk
                                                               ; Edited 20-Mar-2024 11:00 by rmk
                                                               ; Edited 17-Mar-2024 12:06 by rmk
                                                               ; Edited 15-Mar-2024 13:53 by rmk
                                                               ; Edited  6-Jan-2024 15:10 by rmk
                                                               ; Edited 11-Dec-2023 10:02 by rmk
                                                               ; Edited  9-Nov-2023 15:37 by rmk
                                                               ; Edited  5-Nov-2023 11:22 by rmk
                                                               ; Edited 24-Sep-2023 23:30 by rmk
                                                               ; Edited 22-May-2023 22:50 by rmk
                                                               ; Edited 20-May-2023 16:44 by rmk
                                                               ; Edited  8-May-2023 08:44 by rmk
                                                               ; Edited 29-Apr-93 19:47 by jds
```

;; Turn an unformatted TEdit file into a formatted TEdit file, essentially simulating ANY.EOLC by ensuring that end-of-line indicators (CR, CRLF, LF)
;; are canonicalized as EOL's and then interpreted as paragraph ends. Pieces are split so that EOLs are always at piece-end. If it wasn't formatted
;; before, it presumably didn't have any looks to worry about, just the defaults.

;; Using BIN for the main iteration is a little tricky when TEDIT.RPLCHARCODE is used to make the single-character change. RPLCHARCODE
;; can split the pieces and parameters in the TSTREAM that are used to drive the high-speed (BINABLE) operation. It should perhaps figure out
;; how to fix the stream internally, but for now the \TEXTSETFILEPTR gets things consistent again.

```
(LET [(TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM]
     (CL:UNLESS (OR (FGETTOBJ TEXTOBJ FORMATTEDP)
                    (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN)))
         (CL:UNLESS START (SETQ START 1))
         (CL:UNLESS END
             (SETQ END (FGETTOBJ TEXTOBJ TEXTLEN)))
         (CL:WHEN (IGEQ END START)
             [for CHNO CHANGED CRLF from START first
```

;; CHNO is in characters, one more than stream positions

```
                                                 (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 START))
               do (SELCHARQ (BIN TSTREAM)
                        (LF
```

;; Linefeed not preceded by CR, replace by EOL and mark it paragraph-last. What about FORM?

```
                            (TEDIT.RPLCHARCODE TEXTOBJ CHNO (CHARCODE EOL))
                            (SETQ CHANGED T)
                            (FSETPC (\TEDIT.CHTOPC CHNO TEXTOBJ)
```

```
                                PPARALAST T))
                            (CR ;; Post-CR characters go to a separate piece, the CR piece is then paragraph-final
                                (FSETPC (PREVPIECE (\TEDIT.ALIGNEDPIECE (ADD1 CHNO)
                                                                        TEXTOBJ))
                                       PPARALAST T)
                                (CL:WHEN (EQ (CHARCODE LF)
                                             (\TEDIT.TEXTPEEKBIN TSTREAM T))
                                    ;; Linefeed following CR.  Chop it off from whatever follows, and then delete it.
                                    (SETQ CRLF T)
                                    (add END -1)                ; One less char to do
                                    (\TEDIT.DELETEPIECES (\TEDIT.SELPIECES (ADD1 CHNO)
                                                                           (ADD1 CHNO)
                                                                           TEXTOBJ)
                                          TEXTOBJ)
                                    ;; We deleted the LF at CHNO, setting the fileptr there resynchronizes on the character just after it.
                                    (\TEDIT.TEXTSETFILEPTR TSTREAM CHNO))
                                (SETQ CHANGED T))
                            NIL)                                ; Test END explicitly, because it may get reduced

                    repeatuntil (IGEQ CHNO END) finally (FSETTOBJ TEXTOBJ FORMATTEDP T)
                                                        (CL:WHEN CHANGED
                                                            (FSETTOBJ TEXTOBJ \DIRTY T)
                                                            (\TEDIT.UPDATE.LINES (CL:IF CRLF
                                                                                        'DELETION
                                                                                        'CHANGED)
                                                                  START
                                                                  (ADD1 (IDIFFERENCE END START))))])])

)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \TEDIT.TRANSLATE.ASCII.CHARLOOKS MACRO
          [OPENLAMBDA (TEXTOBJ CLOOKS NEWFONTNAME)
            ;; Macro because CLOOKSLIST is set.  The alist avoids creating and then uniquifying each time we want to make the same translation.
            (CDR (OR (ASSOC CLOOKS CLOOKSLIST)
                     (CAR (PUSH CLOOKSLIST (CONS CLOOKS
                                                 (\TEDIT.UNIQUIFY.CHARLOOKS
                                                  (LET ((NEWFONT (\TEDIT.FONTCOPY (GETCLOOKS CLOOKS CLFONT)
                                                                                  (LIST 'FAMILY NEWFONTNAME)
                                                                                  TEXTOBJ)))
                                                       (create CHARLOOKS using CLOOKS CLFONT _ NEWFONT CLNAME _
                                                                               (FONTUNPARSE NEWFONT)))
                                                  TEXTOBJ])
)

(DEFINEQ

(\TEDIT.UNIQUIFY.CHARLOOKS
  [LAMBDA (NEWLOOK TEXTOBJ)                                     ; Edited 16-Mar-2024 00:32 by rmk
                                                                ; Edited 15-Oct-2023 17:17 by rmk
                                                                ; Edited 18-Aug-2023 21:47 by rmk
                                                                ; Edited 15-Aug-2023 20:57 by rmk
                                                                ; Edited  3-Aug-2023 17:52 by rmk
                                                                ; Edited 30-May-91 21:40 by jds
    ;; Assure that there is only ONE of a given CHARLOOKS in the document--so that all instances of that set of looks share structure. When we get a
    ;; hit, we move it to the front, hopefully more frequent looks will come earlier in the list.
    (CL:WHEN NEWLOOK
        (for LOOKTAIL LOOK PREVTAIL on (FGETTOBJ TEXTOBJ TXTCHARLOOKSLIST)
           do (SETQ LOOK (CAR LOOKTAIL))
              (CL:WHEN (\TEDIT.EQCLOOKS NEWLOOK LOOK)
                  (CL:WHEN PREVTAIL                             ; Not already in first position
                      (RPLACD PREVTAIL (CDR LOOKTAIL))
                      (push (FGETTOBJ TEXTOBJ TXTCHARLOOKSLIST)
                            LOOK))
                  (RETURN LOOK))
              (SETQ PREVTAIL LOOKTAIL)
           finally (push (FGETTOBJ TEXTOBJ TXTCHARLOOKSLIST)
                         NEWLOOK)
                   (RETURN NEWLOOK)))])

(\TEDIT.UNIQUIFY.PARALOOKS
  [LAMBDA (NEWLOOK TEXTOBJ)                                     ; Edited 16-Mar-2024 00:30 by rmk
                                                                ; Edited 18-Aug-2023 21:48 by rmk
                                                                ; Edited 30-May-91 21:41 by jds
    ;; Assure that there is only ONE of a given PARALOOKS in the document--so that all instances of that set of looks share structure. When we get a
    ;; hit, we move it to the front, hopefully more frequent looks will come earlier in the list.
    (for LOOKTAIL LOOK PREVTAIL on (GETTOBJ TEXTOBJ TXTPARALOOKSLIST)
       do (SETQ LOOK (CAR LOOKTAIL))
          (CL:WHEN (\TEDIT.EQFMTSPEC NEWLOOK LOOK)
```

```
            (CL:WHEN PREVTAIL                                   ; Not already in first position
                (RPLACD PREVTAIL (CDR LOOKTAIL))
                (push (GETTOBJ TEXTOBJ TXTPARALOOKSLIST)
                      LOOK))
            (RETURN LOOK))
        (SETQ PREVTAIL LOOKTAIL)
    finally (push (GETTOBJ TEXTOBJ TXTPARALOOKSLIST)
                  NEWLOOK)
            (RETURN NEWLOOK])
```

(\TEDIT.UNIQUIFY.ALL

```
  [LAMBDA (TEXTOBJ)                                             ; Edited  8-Feb-2025 20:24 by rmk
                                                                ; Edited 16-Mar-2024 10:03 by rmk
                                                                ; Edited 14-Nov-2023 16:20 by rmk
                                                                ; Edited 25-Aug-2023 08:57 by rmk
                                                                ; Edited 15-Aug-2023 22:04 by rmk
                                                                ; Edited  3-Aug-2023 18:44 by rmk
                                                                ; Edited  1-Aug-2023 11:43 by rmk
                                                                ; Edited 13-Jul-2022 22:56 by rmk
    (SETTOBJ TEXTOBJ TXTCHARLOOKSLIST NIL)
    (SETTOBJ TEXTOBJ TXTPARALOOKSLIST NIL)
    (for PC inpieces (\TEDIT.FIRSTPIECE TEXTOBJ) do             ;; Assure that the CHARLOOKS and PARALOOKS of every piece are in the cache.
                                                                    (change (PLOOKS PC)
                                                                            (\TEDIT.UNIQUIFY.CHARLOOKS DATUM TEXTOBJ))
                                                                    (change (PPARALOOKS PC)
                                                                            (\TEDIT.UNIQUIFY.PARALOOKS DATUM TEXTOBJ)))
    (CL:WHEN (GETTOBJ TEXTOBJ DEFAULTCHARLOOKS)
        (change (GETTOBJ TEXTOBJ DEFAULTCHARLOOKS)
                (\TEDIT.UNIQUIFY.CHARLOOKS DATUM TEXTOBJ)))
    (change (GETTOBJ TEXTOBJ CARETLOOKS)
            (\TEDIT.UNIQUIFY.CHARLOOKS DATUM TEXTOBJ))
    (change (GETTOBJ TEXTOBJ DEFAULTPARALOOKS)
            (\TEDIT.UNIQUIFY.PARALOOKS DATUM TEXTOBJ])
```

(\TEDIT.FLUSH.UNUSED.LOOKS

```
  [LAMBDA (TEXTOBJ)                                             ; Edited 19-Feb-2025 11:56 by rmk
                                                                ; Edited  8-Feb-2025 20:36 by rmk
                                                                ; Edited 16-Mar-2024 10:03 by rmk
                                                                ; Edited 25-Aug-2023 08:03 by rmk
                                                                ; Edited 15-Aug-2023 22:11 by rmk
                                                                ; Edited 30-May-91 21:47 by jds
    ;; Run thru the CHARLOOKS and PARALOOKS lists for this document, and flush any looks that aren't being used in the document itself.
    (LET ((CHARLOOKS (GETTOBJ TEXTOBJ TXTCHARLOOKSLIST))
          (PARALOOKS (GETTOBJ TEXTOBJ TXTPARALOOKSLIST)))
        ;; Reset the in-use mark in all looks
        (for LOOKS in CHARLOOKS do (SETCLOOKS LOOKS CLMARK NIL))
        (for LOOKS in PARALOOKS do (SETPLOOKS LOOKS FMTMARK NIL))
        ;; Run thru the pieces in the document, marking the looks that are really in use.
        (for PC inpieces (\TEDIT.FIRSTPIECE TEXTOBJ) do (FSETCLOOKS (PLOOKS PC)
                                                                    CLMARK T)
                                                         (FSETPLOOKS (PPARALOOKS PC)
                                                                    FMTMARK T))
        ;; Keep only those char and para looks that ARE being used.
        (SETTOBJ TEXTOBJ TXTCHARLOOKSLIST (for LOOKS in CHARLOOKS when (FGETCLOOKS LOOKS CLMARK)
                                             collect (FSETCLOOKS LOOKS CLMARK NIL)
                                                     LOOKS))
        (SETTOBJ TEXTOBJ TXTPARALOOKSLIST (for LOOKS in PARALOOKS when (FGETPLOOKS LOOKS FMTMARK)
                                             collect (FSETPLOOKS LOOKS FMTMARK NIL)
                                                     LOOKS])

)

;; Public entries

(DEFINEQ
```

(TEDIT.LOOKS

```
  [LAMBDA (TSTREAM NEWLOOKS SELORCH# LEN)                       ; Edited 11-Aug-2024 18:11 by rmk
                                                                ; Edited  2-Aug-2024 08:46 by rmk
                                                                ; Edited 27-Jul-2024 23:49 by rmk
                                                                ; Edited 25-Jul-2024 15:01 by rmk
                                                                ; Edited 13-Jul-2024 16:04 by rmk
                                                                ; Edited 22-May-2024 13:55 by rmk
                                                                ; Edited  9-Feb-2024 11:40 by rmk
                                                                ; Edited 23-Dec-2023 14:12 by rmk
                                                                ; Edited 28-May-2023 13:56 by rmk
                                                                ; Edited 24-May-2023 23:12 by rmk
                                                                ; Edited 30-May-91 21:41 by jds
    ;; Programmatic interface for character looks in TEdit.  Applies to the LEN characters starting at SELORCH#, or the characters selected by
    ;; SELORCH# if it is a selection.  Nothing to do if the selection isn't set.  POINT is preserved and used only to set the caret looks.
```

```
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    ;; Ignores LEN if SELORCH# is a selection
    [\TEDIT.CHANGE.CHARLOOKS TSTREAM NEWLOOKS (if (type? SELECTION SELORCH#)
                                                  then SELORCH#
                                                elseif SELORCH#
                                                  then (TEDIT.SETSEL TSTREAM SELORCH# LEN 'LEFT)
                                                else (TEXTSEL (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM]
    ;; Out of bounds or maybe a point selection, no text to change. Punt out after setting the caret looks.  Old code did not set the history, should we?
    (TEDIT.CARETLOOKS TSTREAM NEWLOOKS)
    TSTREAM])
```

(TEDIT.GET.LOOKS

```
  [LAMBDA (TEXTOBJ CH#ORCHARLOOKS)                                  ; Edited 17-Mar-2024 00:27 by rmk
                                                                    ; Edited 14-Dec-2023 21:00 by rmk
                                                                    ; Edited 21-Jun-2023 11:10 by rmk
                                                                    ; Edited 22-Aug-2022 13:14 by rmk
                                                                    ; Edited 30-May-91 21:44 by jds
    ;; Returns as a property list the looks denoted by CH#ORCHARLOOKS.
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (\TEDIT.UNPARSE.CHARLOOKS.LIST (if (type? CHARLOOKS CH#ORCHARLOOKS)
                                       then                         ; Unparse the given looks.
                                            CH#ORCHARLOOKS
                                     elseif (ZEROP (TEXTLEN TEXTOBJ))
                                       then                         ; Empty document, use extant caret looks.
                                            (FGETTOBJ TEXTOBJ CARETLOOKS)
                                     else (PLOOKS (\TEDIT.CHTOPC (OR (FIXP CH#ORCHARLOOKS)
                                                                      (GETSEL (if (type? SELECTION CH#ORCHARLOOKS)
                                                                                  then CH#ORCHARLOOKS
                                                                                elseif (NULL CH#ORCHARLOOKS)
                                                                                  then (TEXTSEL TEXTOBJ)
                                                                                else (\ILLEGAL.ARG CH#ORCHARLOOKS)
                                                                                  )
                                                                              CH#))
                                                  TEXTOBJ])
```

(TEDIT.SUBLOOKS

```
  [LAMBDA (TEXTSTREAM OLDLOOKSLIST NEWLOOKSLIST)                    ; Edited 25-Nov-2024 21:57 by rmk
                                                                    ; Edited  5-Jul-2024 22:54 by rmk
                                                                    ; Edited 25-Jun-2024 11:59 by rmk
                                                                    ; Edited 18-May-2024 16:22 by rmk
                                                                    ; Edited 10-May-2024 22:42 by rmk
                                                                    ; Edited 17-Mar-2024 17:17 by rmk
                                                                    ; Edited  6-May-2024 17:27 by rmk
                                                                    ; Edited 16-Mar-2024 10:03 by rmk
                                                                    ; Edited 13-Nov-2023 00:26 by rmk
                                                                    ; Edited 18-Apr-2023 23:53 by rmk
                                                                    ; Edited 22-Aug-2022 13:06 by rmk
                                                                    ; Edited 26-Apr-93 14:53 by jds
```

;;; User entry to substitute one set of looks for another. Goes through the whole textstream and whenever the looks match the characteristics of
;;; OLDLOOKSLIST which are specified, the characteristics listed in NEWLOOKSLIST are substituted.

```
    (LET ((TEXTOBJ (TEXTOBJ TEXTSTREAM)))                           ; Turn off the selection, first.
         (CL:UNLESS (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
             (for PC CHANGEMADE SEL FIRSTCHANGEDCHNO (NCHARSCHANGED _ 0)
                 (OLDLOOKS _ (\TEDIT.PARSE.CHARLOOKS.LIST OLDLOOKSLIST NIL TEXTOBJ))
                 (NEWLOOKS _ (\TEDIT.PARSE.CHARLOOKS.LIST NEWLOOKSLIST NIL TEXTOBJ))
                 (FEATURELIST _ (for A on OLDLOOKSLIST by (CDDR A) collect (CAR A)))
                 inpieces
                 (\TEDIT.FIRSTPIECE TEXTOBJ) as CH# from 1 by (PLEN PC) when (\TEDIT.SAMECLOOKS OLDLOOKS
                                                                                                 (PLOOKS PC)
                                                                                                 FEATURELIST)
               do (CL:UNLESS CHANGEMADE
                      (SETQ CHANGEMADE T)
                      (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
                      (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
                      (FSETTOBJ TEXTOBJ \DIRTY T))
                  ;; Note that we may be creating new looks each time, depending on what is there and what is changed.
                  (FSETPC PC PLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS (\TEDIT.PARSE.CHARLOOKS.LIST NEWLOOKSLIST
                                                                                             (PLOOKS PC)
                                                                                             TEXTOBJ)
                                                               TEXTOBJ))
                  ;; This goes piece by piece, each one adding to the collection of dirty lines.  We keep track of the first and last changes
                  (CL:UNLESS FIRSTCHANGEDCHNO (SETQ FIRSTCHANGEDCHNO CH#))
                  (add NCHARSCHANGED (PLEN PC))
               finally (CL:WHEN (AND CHANGEMADE (\TEDIT.PRIMARYPANE TEXTOBJ))
                                                                    ; Update the screen image
                             (\TEDIT.UPDATE.LINES TEXTOBJ 'LOOKS FIRSTCHANGEDCHNO NCHARSCHANGED)
                             (\TEDIT.SHOWSEL SEL T TEXTOBJ))
```

```
                    (RETURN CHANGEMADE))])
```

(TEDIT.FINDLOOKS

```
  [LAMBDA (TEXTSTREAM OLDLOOKSLIST CH#)                    ; Edited 17-Mar-2024 00:27 by rmk
                                                           ; Edited  3-Dec-2023 00:09 by rmk
                                                           ; Edited 13-Nov-2023 00:26 by rmk
                                                           ; Edited 18-Apr-2023 23:53 by rmk
                                                           ; Edited 22-Aug-2022 13:06 by rmk
                                                           ; Edited 26-Apr-93 14:53 by jds
```

;;; Finds and selects the next substring of the text whose looks are a superset of OLDLOOKSLIST.

```
    (LET ((TEXTOBJ (TEXTOBJ TEXTSTREAM)))                  ; Turn off the selection, first.
         (if (AND (FIXP CH#)
                  (IGEQ CH# 1)
                  (ILEQ CH# (FGETTOBJ TEXTOBJ TEXTLEN)))
           elseif (type? SELECTION CH#)
             then (SETQ CH# (TEDIT.GETPOINT TEXTOBJ CH#))
           elseif (NULL CH#)
             then (SETQ CH# (TEDIT.GETPOINT TEXTOBJ (FGETTOBJ TEXTOBJ SEL)))
           else (\ILLEGAL.ARG CH#))
         (CL:UNLESS (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
             [for PC PCLAST FOUNDCH# (OLDLOOKS _ (\TEDIT.PARSE.CHARLOOKS.LIST OLDLOOKSLIST NIL TEXTOBJ))
                  (FEATURELIST _ (for A on OLDLOOKSLIST by (CDDR A) collect (CAR A)))
                  inpieces
                  (\TEDIT.CHTOPC CH# TEXTOBJ) when (\TEDIT.SAMECLOOKS OLDLOOKS (PLOOKS PC)
                                                                      FEATURELIST)
                     do [SETQ PCLAST (find PC1 inpieces (NEXTPIECE PC) suchthat (NOT (\TEDIT.SAMECLOOKS OLDLOOKS
                                                                                                    (PLOOKS PC1)
                                                                                                    FEATURELIST]
                        (SETQ PCLAST (CL:IF PCLAST
                                            (PREVPIECE PCLAST)
                                            PC))
                        (SETQ FOUNDCH# (\TEDIT.PCTOCH PC TEXTOBJ))
                        (TEDIT.SETSEL TEXTOBJ FOUNDCH# (IDIFFERENCE (IPLUS (\TEDIT.PCTOCH PCLAST)
                                                                           (PLEN PCLAST))
                                                                    FOUNDCH#)
                               'RIGHT)
                        (TEDIT.NORMALIZECARET TEXTOBJ)
                        (RETURN (\TEDIT.COPYSEL (FGETTOBJ TEXTOBJ SEL])])

)

(RPAQ? TEDIT.FONTCLASSES '(DISPLAY PDF POSTSCRIPT INTERPRESS PRESS))

(DEFINEQ
```

(\TEDIT.CHANGE.CHARLOOKS

```
  [LAMBDA (TSTREAM NEWLOOKS TARGETSEL)                     ; Edited 31-Jan-2025 10:31 by rmk
                                                           ; Edited  1-Jan-2025 18:11 by rmk
                                                           ; Edited 29-Dec-2024 20:08 by rmk
                                                           ; Edited 26-Nov-2024 23:50 by rmk
                                                           ; Edited 22-Oct-2024 23:37 by rmk
                                                           ; Edited  2-Oct-2024 14:22 by rmk
                                                           ; Edited 28-Sep-2024 17:58 by rmk
                                                           ; Edited 16-Aug-2024 22:41 by rmk
                                                           ; Edited 11-Aug-2024 21:12 by rmk
                                                           ; Edited  6-Aug-2024 09:33 by rmk
                                                           ; Edited 31-Jul-2024 12:05 by rmk
                                                           ; Edited 25-Jun-2024 11:59 by rmk
                                                           ; Edited 15-Mar-2024 14:23 by rmk
                                                           ; Edited 23-Dec-2023 15:24 by rmk
                                                           ; Edited 31-Oct-2023 19:40 by rmk
                                                           ; Edited 24-Jul-2023 17:20 by rmk
                                                           ; Edited 28-May-2023 14:38 by rmk
                                                           ; Edited 19-Apr-93 14:08 by jds
```

;;; Internal programmatic interface to changing character looks. DOES NOT CHANGE the current selection (unless it's the TARGETSEL).

```
    (PROG ((TEXTOBJ (TEXTOBJ TSTREAM))
           SELPIECES NEWLOOKSLIST FONT)                    ; Construct the set of new looks to apply:
          (CL:UNLESS TARGETSEL
              (SETQ TARGETSEL (TEXTSEL TEXTOBJ)))
          (CL:UNLESS (AND NEWLOOKS (FGETSEL TARGETSEL SET)
                          (NOT (\TEDIT.READONLY TSTREAM NIL (GETSEL TARGETSEL CH#)))
                          (ILEQ (GETSEL TARGETSEL CH#)
                                (TEXTLEN TEXTOBJ))
                          (IGEQ (GETSEL TARGETSEL CH#)
                                1))
                 (RETURN NIL))
          (if (type? CHARLOOKS NEWLOOKS)
              then (SETQ NEWLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS NEWLOOKS TEXTOBJ))
            elseif (FONTP NEWLOOKS)
              then (SETQ NEWLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS (\TEDIT.CHARLOOKS.FROM.FONT NEWLOOKS)
                                                             TEXTOBJ))
            elseif (for PTAIL on NEWLOOKS by (CDDR PTAIL) unless (OR (\TEDIT.CHARLOOK.FEATUREP (CAR PTAIL))
```

```
                                                                    (NULL (CADR PTAIL)))
                    do ;; OK if a known property or NIL value.  Caller can delete temporary properties.
                       (TEDIT.PROMPTPRINT TSTREAM (CONCAT (CAR PTAIL)
                                                          " is not a valid character property--aborted")
                              T T)
                       (RETURN T))
                 then (RETURN)
               elseif (AND (SETQ FONT (LISTGET NEWLOOKS 'FONT))
                           (for PTAIL on NEWLOOKS by (CDDR PTAIL)
                              when (MEMB (CAR PTAIL)
                                         '(FAMILY FACE SIZE SLOPE WEIGHT BOLD ITALIC EXPANSION))
                              do (TEDIT.PROMPTPRINT TSTREAM (CONCAT "Cannot specify both FONT and font attributes
                                                                    like " (CAR PTAIL)
                                                                    "--aborted")
                                                    T T)
                                 (RETURN T)))
                 then (RETURN))
            (SETQ SELPIECES (\TEDIT.SELPIECES TARGETSEL NIL TEXTOBJ))
    ;; Verify that all of the new looks are OK before we change anything
            [SETQ NEWLOOKSLIST (for PC OLDCHARLOOKS inselpieces SELPIECES
                                  collect (SETQ OLDCHARLOOKS (PLOOKS PC))
                                          (OR (CL:IF (type? CHARLOOKS NEWLOOKS)
                                                     NEWLOOKS
                                                     (\TEDIT.CHANGE.CHARLOOKS.NEW NEWLOOKS OLDCHARLOOKS TEXTOBJ))
                                              (RETURN NIL]
            (CL:UNLESS NEWLOOKSLIST                                 ; At least one bad font?
                (RETURN NIL))
            (for PC UNDOLIST NEWCHARLOOKS DIRTY (FIRSTCHAR _ (GETSPC SELPIECES SPFIRSTCHAR))
                (ORIGFILEPTR _ (\TEDIT.TEXTGETFILEPTR TSTREAM))
                OLDCHARLOOKS inselpieces SELPIECES as NEWCHARLOOKS in NEWLOOKSLIST
              do (SETQ OLDCHARLOOKS (PLOOKS PC))
                 (add FIRSTCHAR (PLEN PC))                          ; Beginning of next piece--where to stop undoing if new pieces
                                                                    ; inserted
                 (if (\TEDIT.EQCLOOKS OLDCHARLOOKS NEWCHARLOOKS)
                     then (SETQ OLDCHARLOOKS NIL)                   ; Undo skips if NIL
                   else (FSETPC PC PLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS NEWCHARLOOKS TEXTOBJ))
                        (CL:UNLESS DIRTY                            ; Resetting DIRTY is expensive, only do it once
                            (FSETTOBJ TEXTOBJ \DIRTY T)
                            (SETQ DIRTY T)))
                 (push UNDOLIST (CONS FIRSTCHAR OLDCHARLOOKS))
              finally ;; Create an event even if no change, so that NEWLOOKS is still available for REDO.
                      [\TEDIT.HISTORYADD TEXTOBJ (\TEDIT.HISTORY.EVENT TEXTOBJ :CharLooks SELPIECES NIL NIL NIL
                                                                       (CONS NEWLOOKS (AND DIRTY (DREVERSE UNDOLIST]
                      (CL:WHEN DIRTY                                ; Something changed
                          (CL:WHEN (\TEDIT.PRIMARYPANE TEXTOBJ)
                              (\TEDIT.SHOWSEL NIL NIL TEXTOBJ)
                              (SELECTQ (LISTGET NEWLOOKS 'INVISIBLE)
                                   (ON ;; Previously visible characters have disappeared, drop the selection to a point
                                       (\TEDIT.UPDATE.SEL (TEXTSEL TEXTOBJ)
                                              0
                                              'LEFT))
                                   (OFF ;; Previously invisible characters have appeared, expand the selection
                                        (\TEDIT.UPDATE.SEL (TEXTSEL TEXTOBJ)
                                               (GETSEL TARGETSEL CH#)
                                               (GETSEL TARGETSEL DCH)
                                               'RIGHT))
                                   NIL)
                              ;; Set caret looks to the looks of the last selected character--the looks of that piece may have been only partially
                              ;; modified
                              (FSETTOBJ TEXTOBJ CARETLOOKS (PLOOKS (\TEDIT.CHTOPC (IMAX 1 (SUB1 (TEDIT.GETPOINT
                                                                                                 TEXTOBJ)))
                                                                                  TEXTOBJ)))
                              (\TEDIT.RESET.EXTEND.PENDING.DELETE TEXTOBJ)
                              (\TEDIT.UPDATE.LINES TEXTOBJ 'LOOKS SELPIECES)
                              (\TEDIT.SHOWSEL NIL T TEXTOBJ)
                              (\TEDIT.TEXTSETFILEPTR TSTREAM ORIGFILEPTR)))])
```

(**TEDIT.CHANGE.CHARLOOKS.NEW**

```
  [LAMBDA (NEWLOOKS OLDCHARLOOKS TEXTOBJ)                           ; Edited  2-Jan-2025 15:49 by rmk
                                                                    ; Edited  1-Jan-2025 09:04 by rmk
                                                                    ; Edited  2-Dec-2024 23:52 by rmk
                                                                    ; Edited 29-Aug-2024 11:12 by rmk
                                                                    ; Edited 22-Aug-2024 10:50 by rmk
                                                                    ; Edited 16-Aug-2024 18:23 by rmk
                                                                    ; Edited 11-Aug-2024 00:12 by rmk
    ;; Make a new CHARLOOKS reflecting the properties in NEWLOOKS, with defaults taken from OLDCHARLOOKS, if given, or the
    ;; DEFAULTCHARLOOKS of TEXTOBJ, if given,;

    ;; OLDCHARLOOKS is also used as the base for increments.Ð

    (CL:UNLESS OLDCHARLOOKS
```

```
        (SETQ OLDCHARLOOKS (OR (AND TEXTOBJ (GETTOBJ TEXTOBJ DEFAULTCHARLOOKS))
                               (create CHARLOOKS))))
    (for NLTAIL NEWFONT VAL NEWCHARLOOKS on NEWLOOKS by (CDDR NLTAIL)
       first (CL:WHEN (EQ 'OFF (LISTGET NEWLOOKS 'DEVICE))
                  (TEDIT.PROMPTPRINT TEXTOBJ "Please specify a particular font device" T)
                  (RETURN NIL))
             (CL:UNLESS (SETQ NEWFONT (\TEDIT.CHARLOOKS.CHANGE.FONT NEWLOOKS OLDCHARLOOKS TEXTOBJ))
                                                                  ; Bad font specification
                  (RETURN NIL))
             (SETQ NEWCHARLOOKS (create CHARLOOKS using OLDCHARLOOKS CLFONT _ NEWFONT CLNAME _ (FONTUNPARSE NEWFONT
                                                                                                          )))
       do (SETQ VAL (CADR NLTAIL))
          (CL:WHEN (MEMB VAL '(NEUTRAL OFF))                      ; Off and NEUTRAL both turn off
              (SETQ VAL NIL))

          ;; Skip the font attributes here, they have already been interpreted

          (SELECTQ (CAR NLTAIL)
               (OVERLINE (FSETCLOOKS NEWCHARLOOKS CLOLINE VAL))
               (SUPERSCRIPT (FSETCLOOKS NEWCHARLOOKS CLOFFSET VAL))
               (SUBSCRIPT (FSETCLOOKS NEWCHARLOOKS CLOFFSET (IMINUS VAL)))
               (PROTECTED (FSETCLOOKS NEWCHARLOOKS CLPROTECTED VAL))
               (UNDERLINE (FSETCLOOKS NEWCHARLOOKS CLULINE VAL))
               (STYLE (FSETCLOOKS NEWCHARLOOKS CLSTYLE VAL))
               (UNBREAKABLE (FSETCLOOKS NEWCHARLOOKS CLUNBREAKABLE VAL))
               (STRIKEOUT (FSETCLOOKS NEWCHARLOOKS CLSTRIKE VAL))
               (INVERTED (FSETCLOOKS NEWCHARLOOKS CLINVERTED VAL))
               ((SELECTPOINT SELAFTER)
                    (FSETCLOOKS NEWCHARLOOKS CLSELAFTER VAL)      ; Mutually exclusive
                    (FSETCLOOKS NEWCHARLOOKS CLSELBEFORE (NOT VAL)))
               (SELBEFORE (FSETCLOOKS NEWCHARLOOKS CLSELBEFORE VAL)
                          (FSETCLOOKS NEWCHARLOOKS CLSELAFTER (NOT VAL)))
               (OFFSETINCREMENT
                    (FSETCLOOKS NEWCHARLOOKS CLOFFSET (IPLUS VAL (OR (AND OLDCHARLOOKS (FGETCLOOKS OLDCHARLOOKS
                                                                                              CLOFFSET))
                                                                     0))))
               (INVISIBLE (FSETCLOOKS NEWCHARLOOKS CLINVISIBLE VAL))
               NIL)
       finally (RETURN NEWCHARLOOKS])
```

(\TEDIT.CHARLOOKS.CHANGE.FONT

```
  [LAMBDA (NEWLOOKS OLDCHARLOOKS TEXTOBJ)                    ; Edited 29-Jan-2025 23:52 by rmk
                                                             ; Edited 10-Jan-2025 11:01 by rmk
                                                             ; Edited  7-Jan-2025 12:34 by rmk
                                                             ; Edited  2-Jan-2025 10:23 by rmk
                                                             ; Edited 29-Dec-2024 20:11 by rmk
                                                             ; Edited 22-Dec-2024 15:27 by rmk
                                                             ; Edited 30-Oct-2024 14:09 by rmk
                                                             ; Edited  7-Sep-2024 13:08 by rmk
                                                             ; Edited 16-Aug-2024 18:25 by rmk
                                                             ; Edited 11-Aug-2024 00:11 by rmk
                                                             ; Edited 30-Jul-2024 22:36 by rmk
                                                             ; Edited 26-Jul-2024 14:55 by rmk
    ;; Converts all the independent font properties into a final list of font properties and uses them to create a new font, with defaults taken from the
    ;; newly specified FONT property, or the font of OLDCHARLOOKS.  Caller guarantees that we don't see both a new FONT and new font attributes.

    ;; Several cases:

    ;;   1.  OLDCHARLOOKCS CLFONT is a FONTDESCRIPTOR, same for all devices

    ;;         If DEVICE is ALL or OFF, CLFONT is replaced by the new FONTDESCRIPTOR that stands for all devices.

    ;;         If DEVICE is a particular device, then CLFONT is coerced to a fontclass that differentiates just for the new device.

    ;;              (If the new font matches the old font, then no need to coerce).

    ;;   2.  Old CLFONT is a FONTCLASS.

    ;;          If DEVICE is ALL or OFF, message and bail.

    ;;          Otherwise, change just the component for that device.

    ;;   3.  If NEWLOOKS contains a fontdescriptor, make sure that no other font attributes are specified.  Then unpack the NEWLOOKS font and
    ;; carry on.

    ;;   4.  If NEWLOOKS contains a fontclass, then if DEVICE is ALL, smash it in.  Otherwise, smash in the component for DEVICE.

    (PROG ((DEVICE (LISTGET NEWLOOKS 'DEVICE))
           (NEWFONT (LISTGET NEWLOOKS 'FONT))
           (NEWFAMILY (LISTGET NEWLOOKS 'FAMILY))
           (FACE (LISTGET NEWLOOKS 'FACE))
           (WEIGHT (LISTGET NEWLOOKS 'WEIGHT))
           (SLOPE (LISTGET NEWLOOKS 'SLOPE))
           (EXPANSION (LISTGET NEWLOOKS 'EXPANSION))
           (SIZE (LISTGET NEWLOOKS 'SIZE))
           (SIZEINCREMENT (LISTGET NEWLOOKS 'SIZEINCREMENT))
           (OLDFONT (FGETCLOOKS OLDCHARLOOKS CLFONT))
           FONTSPEC TEMP)

     ;; If either the new FONT or the old CLFONT are font classes, other properties are not allowed.

          (CL:WHEN NEWFONT
              [SETQ NEWFONT (OR (FONTP NEWFONT)
```

```
                                (FONTCREATE NEWFONT)
                                (PROGN (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT NEWFONT " isn't a valid font
                                                                          descriptor--aborted")
                                                                  T T)
                                       (RETURN])
```

;; NEWFONT is now a font or a font class.

```
      (CL:UNLESS (MEMB NEWFAMILY '(NIL OFF))
          (push FONTSPEC 'FAMILY NEWFAMILY))
      (CL:UNLESS WEIGHT
          (SETQ WEIGHT (SELECTQ (LISTGET NEWLOOKS 'BOLD)
                                (ON 'BOLD)
                                (OFF 'REGULAR)
                                NIL)))
      (CL:UNLESS SLOPE
          (SETQ SLOPE (SELECTQ (LISTGET NEWLOOKS 'ITALIC)
                               (ON 'ITALIC)
                               (OFF 'REGULAR)
                               NIL)))
      (if (OR WEIGHT SLOPE EXPANSION)
          then                                              ; Setting one of these inhibits the FACE parameter
               (CL:IF WEIGHT
                   (push FONTSPEC 'WEIGHT WEIGHT))
               (CL:IF SLOPE
                   (push FONTSPEC 'SLOPE SLOPE))
               (CL:IF EXPANSION
                   (push FONTSPEC 'EXPANSION EXPANSION))
        elseif FACE
          then (push FONTSPEC 'FACE FACE))
      (if SIZE
          then (push FONTSPEC 'SIZE SIZE)
        elseif SIZEINCREMENT
          then ;; If a size increment is specified, then add to the newspecs arg for fontcopy, the entry with the incremented size from the
               ;; current font.
               (push FONTSPEC 'SIZE (IPLUS (FONTPROP (GETCLOOKS OLDCHARLOOKS CLFONT)
                                                     'SIZE)
                                           SIZEINCREMENT)))
      (CL:WHEN (AND NEWFONT FONTSPEC)                       ; Caller should have checked this, but...
          (TEDIT.PROMPTPRINT TEXTOBJ "Cannot specify both FONT and separate font attributes--aborted" T)
          (RETURN))
```

;;

```
      (RETURN (if (EQ 'ALL DEVICE)
                  then (if NEWFONT
                           elseif (type? FONTDESCRIPTOR OLDFONT)
                             then (\TEDIT.FONTCOPY OLDFONT FONTSPEC TEXTOBJ)
                           else ;; We may be changing only some attributes of a multi-device fontclass, can't necessarily collapse to a
                                ;; simple font.
                                (SETQ TEMP (\TEDIT.COERCE.FONTCLASS OLDFONT))
                                                        ; Must be a fontclass, change all components
                                (for D inside TEDIT.FONTDEVICES
                                   do (SETFONTCLASSCOMPONENT TEMP D (\TEDIT.FONTCOPY OLDFONT
                                                                                     `(DEVICE ,D ,@FONTSPEC)
                                                                                     TEXTOBJ)))
                                TEMP)
                       else ;; A specific device
                            (SETQ TEMP (\TEDIT.COERCE.FONTCLASS OLDFONT))
                                                        ; Coerce to a class
                            (SETQ NEWFONT (CL:IF NEWFONT
                                                 (FONTCOPY NEWFONT 'DEVICE DEVICE)
                                                 (\TEDIT.FONTCOPY OLDFONT `(DEVICE ,DEVICE ,@FONTSPEC)
                                                        TEXTOBJ)))
                            (CL:WHEN NEWFONT
                                (SETFONTCLASSCOMPONENT TEMP DEVICE NEWFONT)
                                TEMP)])
```

(\TEDIT.LOOKS

```
  [LAMBDA (TEXTOBJ)                                     ; Edited 28-Jun-2024 21:52 by rmk
                                                        ; Edited 13-Jun-2024 22:10 by rmk
                                                        ; Edited  8-May-2023 21:21 by rmk
                                                        ; Edited 30-May-91 21:41 by jds
```

;; Handler for the middle-button menu's LOOKS button. Brings up 3 menus, for font, face, and size. Then calls TEDIT.LOOKS to make the
;; requested changes.

```
    (RESETLST
        [RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ)
               '(PROGN (\TEDIT.MARKINACTIVE OLDVALUE]
        [LET* ((SEL (GETTOBJ TEXTOBJ SEL))
               (REGION (WINDOWPROP (FGETTOBJ TEXTOBJ PRIMARYPANE)
                                   'REGION))
               (POS (create POSITION
                            XCOORD _ (fetch (REGION LEFT) of REGION)
                            YCOORD _ (fetch (REGION TOP) of REGION)))
```

```
              FONT FACE SIZE NEWLOOKS)
             (CL:WHEN (ILEQ (GETSEL SEL CH#)
                            (TEXTLEN TEXTOBJ))                    ; Otherwise, nothing to change
                 (COND
                    ((FGETSEL SEL SET)
                     (CURSORPOSITION (CREATEPOSITION 0 (fetch HEIGHT of REGION))
                            (FGETTOBJ TEXTOBJ PRIMARYPANE))
                     (SETQ FONT (MENU (create MENU
                                              TITLE _ "Font:"
                                              ITEMS _ (NCONC1 (COPY TEDIT.KNOWN.FONTS)
                                                              (LIST 'Other (LIST (FUNCTION TEDIT.NEW.FONT)
                                                                                 TEXTOBJ)))
                                              CENTERFLG _ T)
                                      POS))                       ; Set the font for the new text.
                     (SETQ FACE (SELECTQ (MENU TEDIT.FACE.MENU POS)
                                         (Bold 'BOLD)
                                         (Italic 'ITALIC)
                                         (Bold% Italic 'BOLDITALIC)
                                         (Regular 'STANDARD)
                                         NIL))                    ; Set the face (bold, etc.)
                     (SETQ SIZE (MENU TEDIT.SIZE.MENU POS))       ; Set the type size
                                                                  ; Construct the set of new looks to apply:
                     (SETQ NEWLOOKS (AND FONT (LIST 'FAMILY FONT)))
                     (CL:WHEN FACE
                         (SETQ NEWLOOKS (CONS 'FACE (CONS FACE NEWLOOKS))))
                     (CL:WHEN SIZE
                         (SETQ NEWLOOKS (CONS 'SIZE (CONS SIZE NEWLOOKS))))
                     (CL:WHEN NEWLOOKS                            ; There's something to do.
                         (TEDIT.LOOKS TEXTOBJ NEWLOOKS SEL)))
                    (T (TEDIT.PROMPTPRINT TEXTOBJ "Please select some text to modify" T)))])])
```

(\TEDIT.FONTCOPY

```
  [LAMBDA (FONT NEWSPECS TEXTOBJ)                                 ; Edited 10-Jan-2025 11:02 by rmk
                                                                  ; Edited 11-Aug-2024 00:01 by rmk
                                                                  ; Edited 22-Feb-2024 15:35 by rmk
                                                                  ; Edited 12-Nov-2023 23:24 by rmk
                                                                  (* jds "26-Dec-84 16:06")
```

;; Cloak FONTCOPY in protection for the user from an unavailable font.

```
    (if (NULL NEWSPECS)
        then                                                      ; No changes specified.  Punt it.
             FONT
      elseif (CAR (NLSETQ (FONTCOPY FONT NEWSPECS)))
      else (LET ((MSG (CONCAT "Can't find font " (OR (LISTGET NEWSPECS 'FAMILY)
                                                     (FONTPROP FONT 'FAMILY))
                              " "
                              (OR (LISTGET NEWSPECS 'SIZE)
                                  (FONTPROP FONT 'SIZE))
                              " "
                              (OR (LISTGET NEWSPECS 'FACE)
                                  (FONTPROP FONT 'FACE))
                              "--aborted")))
                 (if TEXTOBJ
                     then (TEDIT.PROMPTPRINT TEXTOBJ MSG T T)
                   else (ERROR MSG)))
           NIL])
```

(\TEDIT.COERCE.FONTCLASS

```
  [LAMBDA (FONT)                                                  ; Edited  1-Jan-2025 09:11 by rmk
```

;; If FONT is a FONTDESCRIPTOR, returns a new FONTCLASS twith components filled in from FONT. If FONT is a FONTCLASS, returns a
;; TEDIT-specialized copy.

```
    (LET ((NEWCLASS (create FONTCLASS
                            FONTCLASSNAME _ 'TEDIT-FONTCLASS
                            PRETTYFONT# _ 0)))
         [for D in TEDIT.FONTDEVICES do (SETFONTCLASSCOMPONENT NEWCLASS D (CL:IF (type? FONTDESCRIPTOR FONT)
                                                                              (FONTCOPY FONT 'DEVICE D)
                                                                              (FONTCLASSCOMPONENT FONT D))]
         NEWCLASS])

)

;; Paragraph looks functions

(DEFINEQ
```

(\TEDIT.EQFMTSPEC

```
  [LAMBDA (PARALOOK1 PARALOOK2)                                   ; Edited 19-Feb-2025 11:53 by rmk
                                                                  ; Edited  8-Feb-2025 20:43 by rmk
                                                                  ; Edited 28-Jul-2024 21:29 by rmk
                                                                  ; Edited  2-Jul-93 21:32 by sybalskY:MV:ENVOS
```

;; Given two sets of FMTSPECS, are they effectively the same?

```
    (PARALOOKS! PARALOOK1)
    (PARALOOKS! PARALOOK2)
```

```
    (OR (EQ PARALOOK1 PARALOOK2)
        (AND (EQ (FGETPLOOKS PARALOOK1 QUAD)
                 (FGETPLOOKS PARALOOK2 QUAD))
             (EQ (FGETPLOOKS PARALOOK1 FMTNEWPAGEBEFORE)
                 (FGETPLOOKS PARALOOK2 FMTNEWPAGEBEFORE))
             (EQ (FGETPLOOKS PARALOOK1 FMTNEWPAGEAFTER)
                 (FGETPLOOKS PARALOOK2 FMTNEWPAGEAFTER))
             (EQ (FGETPLOOKS PARALOOK1 FMTSTYLE)
                 (FGETPLOOKS PARALOOK2 FMTSTYLE))
             (EQ (FGETPLOOKS PARALOOK1 FMTSPECIALX)
                 (FGETPLOOKS PARALOOK2 FMTSPECIALX))
             (EQ (FGETPLOOKS PARALOOK1 FMTSPECIALY)
                 (FGETPLOOKS PARALOOK2 FMTSPECIALY))
             (EQ (FGETPLOOKS PARALOOK1 FMTHEADINGKEEP)
                 (FGETPLOOKS PARALOOK2 FMTHEADINGKEEP))
             (EQ (FGETPLOOKS PARALOOK1 FMTKEEP)
                 (FGETPLOOKS PARALOOK2 FMTKEEP))
             (EQ (FGETPLOOKS PARALOOK1 FMTPARATYPE)
                 (FGETPLOOKS PARALOOK2 FMTPARATYPE))
             (EQ (FGETPLOOKS PARALOOK1 FMTPARASUBTYPE)
                 (FGETPLOOKS PARALOOK2 FMTPARASUBTYPE))
             (EQ (FGETPLOOKS PARALOOK1 FMTHARDCOPY)
                 (FGETPLOOKS PARALOOK2 FMTHARDCOPY))
             (EQ (FGETPLOOKS PARALOOK1 FMTREVISED)
                 (FGETPLOOKS PARALOOK2 FMTREVISED))
             (EQ (FGETPLOOKS PARALOOK1 FMTCOLUMN)
                 (FGETPLOOKS PARALOOK2 FMTCOLUMN))
             (EQP (FGETPLOOKS PARALOOK1 1STLEFTMAR)
                  (FGETPLOOKS PARALOOK2 1STLEFTMAR))
             (EQP (FGETPLOOKS PARALOOK1 LEFTMAR)
                  (FGETPLOOKS PARALOOK2 LEFTMAR))
             (EQP (FGETPLOOKS PARALOOK1 RIGHTMAR)
                  (FGETPLOOKS PARALOOK2 RIGHTMAR))
             (EQP (FGETPLOOKS PARALOOK1 LEADBEFORE)
                  (FGETPLOOKS PARALOOK2 LEADBEFORE))
             (EQP (FGETPLOOKS PARALOOK1 LEADAFTER)
                  (FGETPLOOKS PARALOOK2 LEADAFTER))
             (EQP (FGETPLOOKS PARALOOK1 LINELEAD)
                  (FGETPLOOKS PARALOOK2 LINELEAD))
             (EQP (FGETPLOOKS PARALOOK1 FMTBASETOBASE)
                  (FGETPLOOKS PARALOOK2 FMTBASETOBASE))
             (EQUAL (FGETPLOOKS PARALOOK1 FMTUSERINFO)
                    (FGETPLOOKS PARALOOK2 FMTUSERINFO))
             (EQUAL (FGETPLOOKS PARALOOK1 FMTCHARSTYLES)
                    (FGETPLOOKS PARALOOK2 FMTCHARSTYLES))
             (EQ (FGETPLOOKS PARALOOK1 FMTDEFAULTTAB)
                 (FGETPLOOKS PARALOOK2 FMTDEFAULTTAB))
             (EQUAL (FGETPLOOKS PARALOOK1 FMTTABS)
                    (FGETPLOOKS PARALOOK2 FMTTABS])
```

(**TEDIT.GET.PARALOOKS**

```
  [LAMBDA (TSTREAM SELORCH#)                                    ; Edited 19-Feb-2025 12:00 by rmk
                                                                ; Edited  8-Feb-2025 20:55 by rmk
                                                                ; Edited  4-Aug-2024 17:17 by rmk
                                                                ; Edited 28-Jul-2024 16:25 by rmk
                                                                ; Edited 17-Mar-2024 00:27 by rmk
                                                                ; Edited 11-Dec-2023 10:12 by rmk
                                                                ; Edited 22-Jun-2023 00:02 by rmk
                                                                ; Edited 11-Feb-2023 14:55 by rmk
                                                                ; Edited 30-May-91 21:44 by jds
    ;; Return a proplist of paragraph formatting information about the characters specified.

    (LET* [(TEXTOBJ (TEXTOBJ TSTREAM))
           (PC (if (type? PIECE SELORCH#)
                   then ;; An internal call, if we already have the piece.
                        SELORCH#
                 else (\TEDIT.CHTOPC (OR (FIXP SELORCH#)
                                         (GETSEL (if (type? SELECTION SELORCH#)
                                                     then SELORCH#
                                                   elseif (NULL SELORCH#)
                                                     then (TEXTSEL TEXTOBJ)
                                                     else (\ILLEGAL.ARG SELORCH#))
                                                 CH#))
                                     TEXTOBJ)))
           (PARALOOKS (CL:IF PC
                             (PPARALOOKS PC)
                             (GETTOBJ TEXTOBJ DEFAULTPARALOOKS))]
          (for PROP in (LIST (FGETPLOOKS PARALOOKS QUAD)
                             (FGETPLOOKS PARALOOKS 1STLEFTMAR)
                             (FGETPLOOKS PARALOOKS LEFTMAR)
                             (FGETPLOOKS PARALOOKS RIGHTMAR)
                             (FGETPLOOKS PARALOOKS LEADBEFORE)
                             (FGETPLOOKS PARALOOKS LEADAFTER)
                             (FGETPLOOKS PARALOOKS LINELEAD)
                             (FGETPLOOKS PARALOOKS FMTBASETOBASE)
```

```
                        (create TABSPEC
                               DEFAULTTAB _ (FGETPLOOKS PARALOOKS FMTDEFAULTTAB)
                               TABS _ (COPY (FGETPLOOKS PARALOOKS FMTTABS)))
                        (FGETPLOOKS PARALOOKS FMTSTYLE)
                        (FGETPLOOKS PARALOOKS FMTCHARSTYLES)
                        (FGETPLOOKS PARALOOKS FMTUSERINFO)
                        (FGETPLOOKS PARALOOKS FMTSPECIALX)
                        (FGETPLOOKS PARALOOKS FMTSPECIALY)
                        (FGETPLOOKS PARALOOKS FMTPARATYPE)
                        (FGETPLOOKS PARALOOKS FMTPARASUBTYPE)
                        (ONOFF (FGETPLOOKS PARALOOKS FMTNEWPAGEBEFORE))
                        (ONOFF (FGETPLOOKS PARALOOKS FMTNEWPAGEAFTER))
                        (ONOFF (FGETPLOOKS PARALOOKS FMTHEADINGKEEP))
                        (FGETPLOOKS PARALOOKS FMTKEEP)
                        (ONOFF (FGETPLOOKS PARALOOKS FMTHARDCOPY))
                        (FGETPLOOKS PARALOOKS FMTREVISED)
                        (FGETPLOOKS PARALOOKS FMTCOLUMN))
            as PROPNAME
            in '(QUAD 1STLEFTMARGIN LEFTMARGIN RIGHTMARGIN PARALEADING POSTPARALEADING LINELEADING BASETOBASE
                      TABS STYLE CHARSTYLES USERINFO SPECIALX SPECIALY TYPE SUBTYPE NEWPAGEBEFORE NEWPAGEAFTER
                      HEADINGKEEP KEEP HARDCOPY REVISED COLUMN)
            join (LIST PROPNAME PROP])
```

(\TEDIT.PARSE.PARALOOKS.LIST

```
  [LAMBDA (NEWLOOKS OLDLOOKS TEXTOBJ)                    ; Edited 19-Feb-2025 11:57 by rmk
                                                         ; Edited  8-Feb-2025 22:27 by rmk
                                                         ; Edited 28-Jul-2024 22:14 by rmk
                                                         ; Edited 29-Apr-2024 11:03 by rmk
                                                         ; Edited 17-Oct-2023 12:08 by rmk
                                                         ; Edited  9-May-2023 13:20 by rmk
                                                         ; Edited  5-Sep-2022 15:39 by rmk
                                                         ; Edited  3-Jul-93 21:49 by sybalskY:MV:ENVOS
                                                         ; Apply a given format spec to the paragraphs which are included
                                                         ; in this guy.
    (if (type? PARALOOKS NEWLOOKS)
        then                                             ; if we were given a PARALOOKS it replace the PARALOOKS of
                                                         ; all pieces affected
            NEWLOOKS
      else (LET (NEWFMT 1STLEFT LEFT RIGHT LEADB LEADA LLEAD TABSPEC QUADD NLOOKSAVE TYPE SUBTYPE TYPESET
                        SUBTYPESET NEWBEFORESET NEWBEFORE NEWAFTERSET NEWAFTER KEEP KEEPSET HEADINGKEEP BASETOBASE
                        BASESET REVISED REVISEDSET COLUMN COLUMNSET USERINFO USERINFOSET SPECIALX SPECXSET
                        SPECIALY SPECYSET STYLE STYLESET CHARSTYLES CHARSTYLESSET DEFTAB TABS)
                                                         ; create PARALOOKS from the Plist
                (SETQ 1STLEFT (LISTGET NEWLOOKS '1STLEFTMARGIN))
                (SETQ LEFT (LISTGET NEWLOOKS 'LEFTMARGIN))
                (SETQ RIGHT (LISTGET NEWLOOKS 'RIGHTMARGIN))
                (SETQ LEADB (LISTGET NEWLOOKS 'PARALEADING))
                (SETQ LEADA (LISTGET NEWLOOKS 'POSTPARALEADING))
                (SETQ LLEAD (LISTGET NEWLOOKS 'LINELEADING))
                (SETQ TYPESET (FMEMB 'TYPE NEWLOOKS))
                (SETQ TYPE (LISTGET NEWLOOKS 'TYPE))
                (SETQ SUBTYPESET (FMEMB 'SUBTYPE NEWLOOKS))
                (SETQ SUBTYPE (LISTGET NEWLOOKS 'SUBTYPE))
                (SETQ NEWBEFORESET (FMEMB 'NEWPAGEBEFORE NEWLOOKS))
                (SETQ NEWBEFORE (LISTGET NEWLOOKS 'NEWPAGEBEFORE))
                (SETQ NEWAFTERSET (FMEMB 'NEWPAGEAFTER NEWLOOKS))
                (SETQ NEWAFTER (LISTGET NEWLOOKS 'NEWPAGEAFTER))
                (SETQ HEADINGKEEP (LISTGET NEWLOOKS 'HEADINGKEEP))
                                                         ; Keep for headings
                (SETQ KEEP (LISTGET NEWLOOKS 'KEEP))     ; More general 'Keep-together' spec -- undefined as of 5/22/85
                (SETQ KEEPSET (FMEMB 'KEEP NEWLOOKS))
                (SETQ BASETOBASE (LISTGET NEWLOOKS 'BASETOBASE))
                (SETQ BASESET (FMEMB 'BASETOBASE NEWLOOKS))
                (SETQ REVISED (LISTGET NEWLOOKS 'REVISED))
                (SETQ REVISEDSET (FMEMB 'REVISED NEWLOOKS))
                (SETQ QUADD (LISTGET NEWLOOKS 'QUAD))
                (SETQ COLUMN (LISTGET NEWLOOKS 'COLUMN))
                (SETQ COLUMNSET (FMEMB 'COLUMN NEWLOOKS))
                (SETQ USERINFO (LISTGET NEWLOOKS 'USERINFO))
                (SETQ USERINFOSET (FMEMB 'USERINFO NEWLOOKS))
                (SETQ SPECIALX (LISTGET NEWLOOKS 'SPECIALY))
                (SETQ SPECXSET (FMEMB 'SPECIALY NEWLOOKS))
                (SETQ SPECIALY (LISTGET NEWLOOKS 'SPECIALY))
                (SETQ SPECYSET (FMEMB 'SPECIALY NEWLOOKS))
                (SETQ STYLE (LISTGET NEWLOOKS 'STYLE))
                (SETQ STYLESET (FMEMB 'STYLE NEWLOOKS))
                (SETQ CHARSTYLES (LISTGET NEWLOOKS 'CHARSTYLES))
                (SETQ CHARSTYLESSET (FMEMB 'CHARSTYLES NEWLOOKS))
                (SETQ DEFTAB (LISTGET NEWLOOKS 'DEFAULTTAB))
                (SETQ TABS (LISTGET NEWLOOKS 'TABS))
                (SETQ TABSPEC (LISTGET NEWLOOKS 'TABSPEC))
                (CL:WHEN TABSPEC
                    (SETQ DEFTAB (fetch (TABSPEC DEFAULTTAB) of TABSPEC))
                    (SETQ TABS (fetch (TABSPEC TABS) of TABSPEC)))
                [SELECTQ QUADD
                     ((LEFT RIGHT CENTERED JUSTIFIED NIL)          ; Do nothing -- we got a valid justification spec
```

```
                )
            ((JUST J)
                (SETQ QUADD 'JUSTIFIED))
            (L (SETQQ QUADD LEFT))
            (R (SETQQ QUADD RIGHT))
            ((C CENTER)
                (SETQQ QUADD CENTERED))
            (PROGN                                          ; We got an illegal QUAD value.  Use LEFT.
                   (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "Illegal paragraph quad " QUADD ", replaced with
                                                      LEFT.")
                          T)
                   (SETQ QUADD 'LEFT]
```

;; change from the users list to the real tabspec, a CONS pair of default width and LIST of TAB record instances

```
(SETQ NEWFMT (create PARALOOKS using (OR OLDLOOKS TEDIT.DEFAULT.FMTSPEC)))
(AND 1STLEFT (FSETPLOOKS NEWFMT 1STLEFTMAR 1STLEFT))
(AND LEFT (FSETPLOOKS NEWFMT LEFTMAR LEFT))
(AND RIGHT (FSETPLOOKS NEWFMT RIGHTMAR RIGHT))
(AND LEADB (FSETPLOOKS NEWFMT LEADBEFORE LEADB))
(AND LEADA (FSETPLOOKS NEWFMT LEADAFTER LEADA))
(AND LLEAD (FSETPLOOKS NEWFMT LINELEAD LLEAD))
(AND TABS (FSETPLOOKS NEWFMT FMTTABS TABS))
(AND DEFTAB (FSETPLOOKS NEWFMT FMTDEFAULTTAB DEFTAB))
(AND QUADD (FSETPLOOKS NEWFMT QUAD QUADD))
(AND TYPESET (FSETPLOOKS NEWFMT FMTPARATYPE TYPE))
(AND SUBTYPESET (FSETPLOOKS NEWFMT FMTPARASUBTYPE SUBTYPE))
(AND NEWBEFORESET (FSETPLOOKS NEWFMT FMTNEWPAGEBEFORE NEWBEFORE))
(AND NEWAFTERSET (FSETPLOOKS NEWFMT FMTNEWPAGEAFTER NEWAFTER))
[AND HEADINGKEEP (FSETPLOOKS NEWFMT FMTHEADINGKEEP (EQ HEADINGKEEP 'ON]
(AND KEEPSET (FSETPLOOKS NEWFMT FMTKEEP KEEP))
(AND BASESET (FSETPLOOKS NEWFMT FMTBASETOBASE BASETOBASE))
(AND REVISEDSET (FSETPLOOKS NEWFMT FMTREVISED REVISED))
(AND COLUMNSET (FSETPLOOKS NEWFMT FMTCOLUMN COLUMN))
(AND SPECXSET (FSETPLOOKS NEWFMT FMTSPECIALX SPECIALX))
(AND SPECYSET (FSETPLOOKS NEWFMT FMTSPECIALY SPECIALY))
(AND STYLESET (FSETPLOOKS NEWFMT FMTSTYLE STYLE))
(AND CHARSTYLESSET (FSETPLOOKS NEWFMT FMTCHARSTYLES CHARSTYLES))
(AND USERINFOSET (FSETPLOOKS NEWFMT FMTUSERINFO USERINFO))
NEWFMT])
```

(**TEDIT.PARALOOKS**

```
[LAMBDA (TSTREAM NEWLOOKS SELORCH# LEN)                 ; Edited 10-Aug-2024 00:23 by rmk
                                                        ; Edited 13-Jul-2024 23:16 by rmk
  (SETQ TSTREAM (TEXTSTREAM TSTREAM))
  (LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
        TARGETSEL)
```

;; Ignores LEN if SELORCH# is a selection

```
       (SETQ TARGETSEL (if (type? SELECTION SELORCH#)
                           then SELORCH#
                         elseif SELORCH#
                           then (TEDIT.SETSEL TSTREAM SELORCH# LEN 'RIGHT)
                         else (TEXTSEL TEXTOBJ)))
       (CL:WHEN (GETSEL TARGETSEL SET)
           (if (\TEDIT.READONLY TEXTOBJ NIL (GETSEL TARGETSEL CH#))
             elseif (AND (ILEQ (GETSEL TARGETSEL CH#)
                               (TEXTLEN TEXTOBJ)))
               then (\TEDIT.CHANGE.PARALOOKS TSTREAM NEWLOOKS TARGETSEL)))])
```

(**\TEDIT.CHANGE.PARALOOKS**

```
[LAMBDA (TSTREAM NEWLOOKS TARGETSEL)                    ; Edited  8-Feb-2025 22:30 by rmk
                                                        ; Edited 31-Jan-2025 09:45 by rmk
                                                        ; Edited  6-Jan-2025 23:41 by rmk
                                                        ; Edited  5-Jan-2025 16:01 by rmk
                                                        ; Edited 26-Nov-2024 23:51 by rmk
                                                        ; Edited 27-Sep-2024 16:06 by rmk
                                                        ; Edited 16-Aug-2024 14:21 by rmk
                                                        ; Edited 11-Aug-2024 21:59 by rmk
                                                        ; Edited  4-Aug-2024 23:19 by rmk
                                                        ; Edited  2-Aug-2024 00:39 by rmk
                                                        ; Edited  1-Aug-2024 00:12 by rmk
                                                        ; Edited 29-Jul-2024 11:20 by rmk
                                                        ; Edited 26-Jul-2024 16:17 by rmk
                                                        ; Edited 13-Jul-2024 22:55 by rmk
```

;; Apply new looks to the piece that begins the paragraph containing the first selected character, the piece that ends the paragraph containing the ;; last piece of the selection, and all pieces in between. All the pieces within a paragraph have the same looks.

;; If we are given a PARALOOKS we replace the PARALOOKS of all pieces in all selected paragraphs. Otherwise, we just override particular ;; values in the selected-paragraph looks.

```
  (PROG ((TEXTOBJ (TEXTOBJ TSTREAM))
         (PROPNAMES '(1STLEFTMARGIN LEFTMARGIN RIGHTMARGIN PARALEADING POSTPARALEADING LINELEADING BASETOBASE
                                    QUAD TYPE SUBTYPE SPECIALX SPECIALY NEWPAGEBEFORE NEWPAGEAFTER HEADINGKEEP KEEP
                                    HARDCOPY USERINFO REVISED STYLE CHARSTYLES COLUMN TABS DEFAULTTAB MARGINBAR))
         PARAPIECES)
```

```
        (CL:UNLESS TARGETSEL
            (SETQ TARGETSEL (TEXTSEL TEXTOBJ)))
        (CL:UNLESS (AND NEWLOOKS (FGETSEL TARGETSEL SET)
                        (NOT (\TEDIT.READONLY TEXTOBJ NIL (GETSEL TARGETSEL CH#)))
                        (ILEQ (GETSEL TARGETSEL CH#)
                              (TEXTLEN TEXTOBJ))
                        (IGEQ (GETSEL TARGETSEL CH#)
                              1))
            (RETURN NIL))
        (if (type? PARALOOKS NEWLOOKS)
            then (SETQ NEWLOOKS (\TEDIT.UNIQUIFY.PARALOOKS NEWLOOKS TEXTOBJ))
          elseif (for PTAIL on NEWLOOKS by (CDDR PTAIL) unless (OR (MEMB (CAR PTAIL)
                                                                         PROPNAMES)
                                                                   (NULL (CADR PTAIL)))
                    do ;; Caller can set NIL to delete temporary properties
                       (TEDIT.PROMPTPRINT TSTREAM (CONCAT (CAR PTAIL)
                                                          " is not a valid paragraph property--aborted")
                              T T)
                       (RETURN T))
             then (RETURN))
    ;;
        (SETQ PARAPIECES (\TEDIT.PARAPIECES TARGETSEL NIL TEXTOBJ))
    ;; Testing OLDPARALOOKS will typically avoid repeated calculation of the same NEWPARALOOKS, given the uniquifying.
    ;; For each changed paragraph we keep track of its first character number and its prior looks, for history. That's because the number of pieces in a
    ;; paragraph may change by the doing and doing of future actions, but their character positions will be restored if undoing gets back to this event.
    ;; No need to record prior looks for unchanged pieces.
        (for PC NEWPARALOOKS OLDPARALOOKS UNDOLIST (FIRSTPARAPIECE _ T)
            (FIRSTCHAR _ (GETSPC PARAPIECES SPFIRSTCHAR))
            (ORIGFILEPTR _ (\TEDIT.TEXTGETFILEPTR TSTREAM))
            inselpieces PARAPIECES do (CL:WHEN FIRSTPARAPIECE
                                       ;; First piece of a new paragraph, get the NEWFMTSPEC for all its pieces
                                       (CL:UNLESS UNDOLIST (\TEDIT.SHOWSEL NIL NIL TEXTOBJ))
                                       (SETQ OLDPARALOOKS (PPARALOOKS PC))
                                       (SETQ NEWPARALOOKS (CL:IF (type? PARALOOKS NEWLOOKS)
                                                                 NEWLOOKS
                                                                 (\TEDIT.CHANGE.PARALOOKS.NEW NEWLOOKS
                                                                                              OLDPARALOOKS TEXTOBJ)))
                                       (CL:UNLESS (\TEDIT.EQFMTSPEC OLDPARALOOKS NEWPARALOOKS)
                                                  ; Something changed
                                           (SETQ NEWPARALOOKS (\TEDIT.UNIQUIFY.PARALOOKS NEWPARALOOKS
                                                                                         TEXTOBJ))
                                           (CL:UNLESS (AND UNDOLIST (LISTP NEWLOOKS)
                                                           (EQ 'HARDCOPY (CAR NEWLOOKS))
                                                           (NULL (CDDR NEWLOOKS)))
                                               ;; Resetting DIRTY is expensive, only do it once.  The document is "dirty" for the titlebar and saving
                                               ;; only if something other than hardcopy-display mode was changed
                                               (FSETTOBJ TEXTOBJ \DIRTY T))
                                           (push UNDOLIST (CONS FIRSTCHAR OLDPARALOOKS))))
                                       (FSETPC PC PPARALOOKS NEWPARALOOKS)
                                       (add FIRSTCHAR (PLEN PC))
                                       (SETQ FIRSTPARAPIECE (PPARALAST PC))
             finally ;; Create an event even if UNDOLIST is NIL, so that NEWLOOKS is still available for REDO.
                     [\TEDIT.HISTORYADD TEXTOBJ (\TEDIT.HISTORY.EVENT TEXTOBJ :ParaLooks PARAPIECES NIL NIL NIL
                                                                      (CONS NEWLOOKS (DREVERSE UNDOLIST]
                     (CL:WHEN UNDOLIST
                         ;; Something changed, update any visible lines.
                         (CL:WHEN (\TEDIT.PRIMARYPANE TEXTOBJ)
                             (\TEDIT.RESET.EXTEND.PENDING.DELETE TEXTOBJ)
                             (\TEDIT.UPDATE.LINES TEXTOBJ 'LOOKS PARAPIECES)
                                                 ; Update the screen image, showing the original selection
                             (\TEDIT.SHOWSEL NIL T TEXTOBJ)))
                     (\TEDIT.TEXTSETFILEPTR TSTREAM ORIGFILEPTR])
```

(**TEDIT.CHANGE.PARALOOKS.NEW**

```
  [LAMBDA (NEWLOOKS OLDPARALOOKS TEXTOBJ)                    ; Edited 19-Feb-2025 11:57 by rmk
                                                             ; Edited  8-Feb-2025 22:31 by rmk
                                                             ; Edited  5-Jan-2025 16:02 by rmk
                                                             ; Edited 31-Aug-2024 15:00 by rmk
                                                             ; Edited 29-Aug-2024 11:13 by rmk
                                                             ; Edited 23-Aug-2024 23:41 by rmk
                                                             ; Edited 11-Aug-2024 21:22 by rmk
    ;; Make a new PARALOOKS reflecting the properties in NEWLOOKS, with defaults taken from OLDPARALOOKS, if given, or the
    ;; DEFAULTPARALOOKS of TEXTOBJ, if given,;

    ;; OLDPARALOOKS is also used as the base for increments.Ð
    (CL:UNLESS OLDPARALOOKS
        (CL:WHEN TEXTOBJ
            (SETQ OLDPARALOOKS (GETTOBJ TEXTOBJ DEFAULTPARALOOKS))))
```

```
    (for NLTAIL VAL NEWPARALOOKS on NEWLOOKS by (CDDR NLTAIL) first (SETQ NEWPARALOOKS (CL:IF OLDPARALOOKS
                                                                                          (create PARALOOKS
                                                                                             using OLDPARALOOKS)
                                                                                          (create PARALOOKS)))
       do (SETQ VAL (CADR NLTAIL))
          (CL:WHEN (MEMB VAL '(NEUTRAL OFF))                    ; NEUTRAL and OFF both turn off the flag
              (SETQ VAL NIL))
          (SELECTQ (CAR NLTAIL)
               (1STLEFTMARGIN
                    (FSETPLOOKS NEWPARALOOKS 1STLEFTMAR VAL))
               (LEFTMARGIN (FSETPLOOKS NEWPARALOOKS LEFTMAR VAL))
               (RIGHTMARGIN (FSETPLOOKS NEWPARALOOKS RIGHTMAR VAL))
               (PARALEADING (FSETPLOOKS NEWPARALOOKS LEADBEFORE VAL))
               (POSTPARALEADING
                    (FSETPLOOKS NEWPARALOOKS LEADAFTER VAL))
               (LINELEADING (FSETPLOOKS NEWPARALOOKS LINELEAD VAL))
               (BASETOBASE (FSETPLOOKS NEWPARALOOKS FMTBASETOBASE VAL))
               (QUAD (CL:WHEN VAL
                         (FSETPLOOKS NEWPARALOOKS QUAD (U-CASE VAL))))
               (TYPE (FSETPLOOKS NEWPARALOOKS FMTPARATYPE (CL:IF (EQ VAL 'ON)
                                                                 'PAGEHEADING)))
               (SUBTYPE (FSETPLOOKS NEWPARALOOKS FMTPARASUBTYPE VAL))
               (SPECIALX (FSETPLOOKS NEWPARALOOKS FMTSPECIALX VAL))
               (SPECIALY (FSETPLOOKS NEWPARALOOKS FMTSPECIALY VAL))
               (NEWPAGEBEFORE
                    (FSETPLOOKS NEWPARALOOKS FMTNEWPAGEBEFORE VAL))
               (NEWPAGEAFTER (FSETPLOOKS NEWPARALOOKS FMTNEWPAGEAFTER VAL))
               (HEADINGKEEP (FSETPLOOKS NEWPARALOOKS FMTHEADINGKEEP VAL))
               (KEEP (FSETPLOOKS NEWPARALOOKS FMTKEEP VAL))
               (HARDCOPY (FSETPLOOKS NEWPARALOOKS FMTHARDCOPY VAL))
               (USERINFO (FSETPLOOKS NEWPARALOOKS FMTUSERINFO VAL))
               (REVISED (FSETPLOOKS NEWPARALOOKS FMTREVISED VAL))
               (STYLE (FSETPLOOKS NEWPARALOOKS FMTSTYLE VAL))
               (CHARSTYLES (FSETPLOOKS NEWPARALOOKS FMTCHARSTYLES VAL))
               (COLUMN (FSETPLOOKS NEWPARALOOKS FMTCOLUMN VAL))
               (TABS [if (LISTP (CAR VAL))
                         then (FSETPLOOKS NEWPARALOOKS FMTTABS VAL)
                       else                                     ; Could be the old (DEF . TABS) format
                            (FSETPLOOKS NEWPARALOOKS FMTTABS (CDR VAL))
                            (CL:WHEN (CAR VAL)
                                (FSETPLOOKS NEWPARALOOKS FMTDEFAULTTAB (CAR VAL)))])
               (DEFAULTTAB (FSETPLOOKS NEWPARALOOKS FMTDEFAULTTAB VAL))
               NIL)
       finally (RETURN NEWPARALOOKS])
```

(**TEDIT.COPY.PARALOOKS**

```
  [LAMBDA (TSTREAM SOURCE DEST)                                 ; Edited 25-Nov-2024 14:43 by rmk
                                                                ; Edited 13-Jul-2024 23:22 by rmk
                                                                ; Edited 29-Apr-2024 12:58 by rmk
                                                                ; Edited 17-Mar-2024 00:27 by rmk
                                                                ; Edited  9-Feb-2024 11:39 by rmk
                                                                ; Edited 18-Apr-2023 23:53 by rmk
                                                                ; Edited 22-Oct-2022 15:29 by rmk
                                                                ; Edited 22-Aug-2022 13:15 by rmk
                                                                ; Edited 30-May-91 21:44 by jds
    ;; Copy the PARAGRAPH LOOKS from one place to another

    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    (PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
           SOURCESTREAM LOOKS DESTOBJ)                          ; get the paragraph looks of the first character of SOURCE
          (if (type? SELECTION SOURCE)
              then (SETQ SOURCESTREAM (OR (GETSEL SOURCE SELTEXTSTREAM)
                                          TSTREAM))
            elseif (FIXP SOURCE)
              then (SETQ SOURCESTREAM TSTREAM)
                   (SETQ SOURCE (\TEDIT.UPDATE.SEL (\TEDIT.COPYSEL (TEXTSEL TEXTOBJ))
                                    SOURCE 1))
            else (\ILLEGAL.ARG SOURCE))
          (if (type? SELECTION DEST)
              then                                              ; make sure that the destination selection is in this document;
                   (CL:UNLESS (OR (EQ TSTREAM (FGETSEL DEST SELTEXTSTREAM))
                                  (NULL (FGETSEL DEST SELTEXTSTREAM)))
                           (\LISPERROR "Destination selection is not in stream " TSTREAM))
            elseif (FIXP DEST)
              then (SETQ DEST (\TEDIT.UPDATE.SEL (\TEDIT.COPYSEL (TEXTSEL TEXTOBJ))
                                 DEST 1))
            else (\ILLEGAL.ARG DEST))
          (\TEDIT.CHANGE.PARALOOKS TSTREAM (PPARALOOKS (\TEDIT.CHTOPC (GETSEL SOURCE CH#)
                                                                      SOURCESTREAM))
                DEST])
```

(**\TEDIT.PARABOUNDS**

```
  [LAMBDA (TEXTOBJ CH#)                                         ; Edited 17-Mar-2024 00:27 by rmk
                                                                ; Edited 26-Mar-2023 12:54 by rmk
                                                                ; Edited 20-Feb-2023 13:55 by rmk
```

```
                                                                    ; Edited 25-Oct-2022 14:50 by rmk
                                                                    ; Edited 22-Aug-2022 13:17 by rmk
                                                                    ; Edited 21-Apr-93 18:22 by jds
    ;; Returns the first and last character number of the paragraph that brackets CH#
    (if (ZEROP (TEXTLEN TEXTOBJ))
        then                                                        ; Empty document
             (CONS 0 0)
      else (LET (CHPIECE START-OF-PIECE START END)
                (DECLARE (SPECVARS START-OF-PIECE))
                (SETQ CHPIECE (\TEDIT.CHTOPC (IMIN CH# (TEXTLEN TEXTOBJ))
                                   TEXTOBJ T))
                (SETQ START START-OF-PIECE)                         ; Find the paragraph's first char
                [for PC backpieces (PREVPIECE CHPIECE) until (PPARALAST PC) do (add START (IMINUS (PLEN PC]
                (SETQ END (SUB1 START-OF-PIECE))                    ; Find the paragraph's last char
                (for PC inpieces CHPIECE do (add END (PLEN PC)) repeatuntil (PPARALAST PC))
                (CONS START END])

)

;; For making paragraph-looks substitutions.

(DEFINEQ

(TEDIT.SUBPARALOOKS
  [LAMBDA (TEXTSTREAM OLDLOOKSLIST NEWLOOKSLIST)                    ; Edited 25-Nov-2024 22:00 by rmk
                                                                    ; Edited  5-Jul-2024 22:54 by rmk
                                                                    ; Edited 25-Jun-2024 11:59 by rmk
                                                                    ; Edited 18-May-2024 16:22 by rmk
                                                                    ; Edited 10-May-2024 22:42 by rmk
                                                                    ; Edited 29-Apr-2024 11:06 by rmk
                                                                    ; Edited  6-May-2024 17:28 by rmk
                                                                    ; Edited 17-Mar-2024 00:27 by rmk
                                                                    ; Edited 15-Mar-2024 14:23 by rmk
                                                                    ; Edited 18-Apr-2023 23:54 by rmk
                                                                    ; Edited 22-Aug-2022 13:13 by rmk
                                                                    ; Edited 26-Apr-93 15:13 by jds

;;; User entry to substitute one set of looks for another.  Goes through the whole textstream and whenever the looks match the characteristics of
;;; OLDLOOKSLIST which are specified, the characteristics listed in NEWLOOKSLIST are substituted.

    (LET ((TEXTOBJ (TEXTOBJ TEXTSTREAM)))
         (for PC CHANGEMADE SEL FIRSTCHANGEDCHNO (NCHARSCHANGED _ 0)
             (OLDLOOKS _ (\TEDIT.PARSE.PARALOOKS.LIST OLDLOOKSLIST))
             (NEWLOOKS _ (\TEDIT.PARSE.PARALOOKS.LIST NEWLOOKSLIST))
             (FEATURELIST _ (for A on OLDLOOKSLIST by (CDDR A) collect (CAR A)))
             inpieces
             (\TEDIT.FIRSTPIECE TEXTOBJ) as CH# from 1 by (PLEN PC) when (SAMEPARALOOKS OLDLOOKS
                                                                                 (PPARALOOKS PC PPARALOOKS)
                                                                                 FEATURELIST)
           do (CL:UNLESS CHANGEMADE                                 ; First change, turn off the selection
                  (SETQ CHANGEMADE T)
                  (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
                  (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
                  (FSETTOBJ TEXTOBJ \DIRTY T))
              (FSETPC PC PPARALOOKS (\TEDIT.UNIQUIFY.PARALOOKS (\TEDIT.PARSE.PARALOOKS.LIST NEWLOOKSLIST
                                                                       (PPARALOOKS PC)
                                                                       TEXTOBJ)
                                                                TEXTOBJ))
              ;; This goes piece by piece, each one adding to the collection of dirty lines.  We keep track of the first and last changes
              (CL:UNLESS FIRSTCHANGEDCHNO (SETQ FIRSTCHANGEDCHNO CH#))
              (add NCHARSCHANGED (PLEN PC))
           finally (CL:WHEN (AND CHANGEMADE (\TEDIT.PRIMARYPANE TEXTOBJ))
                                                                    ; Update the screen image
                       (\TEDIT.UPDATE.LINES TEXTOBJ 'LOOKS FIRSTCHANGEDCHNO NCHARSCHANGED)
                       (\TEDIT.SHOWSEL SEL T TEXTOBJ))
                   (RETURN CHANGEMADE])


(SAMEPARALOOKS
  [LAMBDA (PARALOOKS1 PARALOOKS2 FEATURES)                          ; Edited 19-Feb-2025 11:58 by rmk
                                                                    ; Edited  8-Feb-2025 20:49 by rmk
                                                                    ; Edited 29-Jul-2024 23:34 by rmk
                                                                    ; Edited 28-Jul-2024 16:27 by rmk
                                                                    ; Edited  8-Dec-92 00:44 by jds
    ;; Predicate to determine if CLOOK1 and CLOOK2 are the same in all the characteristics listed in FEATURES
    (PARALOOKS! PARALOOKS1)
    (PARALOOKS! PARALOOKS2)
    (for F in FEATURES always (SELECTQ F
                                  (LEFTMARGIN (IEQP (FGETPLOOKS PARALOOKS1 LEFTMAR)
                                                    (FGETPLOOKS PARALOOKS2 LEFTMAR)))
                                  (1STLEFTMARGIN
                                       (IEQP (FGETPLOOKS PARALOOKS1 1STLEFTMAR)
                                             (FGETPLOOKS PARALOOKS2 1STLEFTMAR)))
                                  (RIGHTMARGIN (IEQP (FGETPLOOKS PARALOOKS1 RIGHTMAR)
```

```
                                        (FGETPLOOKS PARALOOKS2 RIGHTMAR)))
                                (QUAD (EQ (FGETPLOOKS PARALOOKS1 QUAD)
                                          (FGETPLOOKS PARALOOKS2 QUAD)))
                                (POSTPARALEADING
                                     (IEQP (FGETPLOOKS PARALOOKS1 LEADBEFORE)
                                           (FGETPLOOKS PARALOOKS2 LEADBEFORE)))
                                (PARALEADING (IEQP (FGETPLOOKS PARALOOKS1 LEADBEFORE)
                                                   (FGETPLOOKS PARALOOKS2 LEADBEFORE)))
                                (LINELEADING (IEQP (FGETPLOOKS PARALOOKS1 LINELEAD)
                                                   (FGETPLOOKS PARALOOKS2 LINELEAD)))
                                (DEFAULTTAB (EQ (FGETPLOOKS PARALOOKS1 FMTDEFAULTTAB)
                                                (FGETPLOOKS PARALOOKS2 FMTDEFAULTTAB)))
                                (TABS (EQUAL (FGETPLOOKS PARALOOKS1 FMTTABS)
                                             (FGETPLOOKS PARALOOKS2 FMTTABS)))
                                (NEWPAGEBEFORE
                                     (EQ (FGETPLOOKS PARALOOKS1 FMTNEWPAGEBEFORE)
                                         (FGETPLOOKS PARALOOKS2 FMTNEWPAGEBEFORE)))
                                (NEWPAGEAFTER (EQ (FGETPLOOKS PARALOOKS1 FMTNEWPAGEAFTER)
                                                  (FGETPLOOKS PARALOOKS2 FMTNEWPAGEAFTER)))
                                (SPECIALX (IEQP (FGETPLOOKS PARALOOKS1 FMTSPECIALX)
                                                (FGETPLOOKS PARALOOKS2 FMTSPECIALX)))
                                (SPECIALY (IEQP (FGETPLOOKS PARALOOKS1 FMTSPECIALY)
                                                (FGETPLOOKS PARALOOKS2 FMTSPECIALY)))
                                (HEADINGKEEP (EQ (FGETPLOOKS PARALOOKS1 FMTHEADINGKEEP)
                                                 (FGETPLOOKS PARALOOKS2 FMTHEADINGKEEP)))
                                (STYLE (EQUAL (FGETPLOOKS PARALOOKS1 FMTSTYLE)
                                              (FGETPLOOKS PARALOOKS2 FMTSTYLE)))
                                (\TEDIT.THELP (CONCAT F " is an unknown feature of paragraph looks"])

)

(DEFINEQ

(\TEDIT.MARK.REVISION
  [LAMBDA (TEXTOBJ PARALOOKS IMAGESTREAM LINE)                    ; Edited  8-Feb-2025 20:49 by rmk
                                                                  ; Edited 27-May-2023 12:12 by rmk
                                                                  ; Edited 30-May-91 21:38 by jds
    (LET ((SCALE (DSPSCALE NIL IMAGESTREAM)))
         (BLTSHADE BLACKSHADE IMAGESTREAM (IPLUS (GETLD LINE RIGHTMARGIN LINE)
                                                 (FIXR (ITIMES 12 SCALE)))
                (GETLD LINE YBOT)
                (FIXR SCALE)
                (GETLD LINE LHEIGHT)
                'PAINT])

)
```

;; Revision-mark support

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA )
)
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

RECORD INDEX