

File created: 5-Apr-2024 08:01:17 {WMEDLEY}<library>tedit>TEDIT-PAGE.;175

edit by: rmk

changes to: (FNS TEDIT.FORMAT.HARDCOPY)

previous date: 4-Apr-2024 23:17:31 {WMEDLEY}<library>tedit>TEDIT-PAGE.;174

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **TEDIT-PAGECOMS**

```
((DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS TEDITPAPERSIZE)
  (EXPORT (RECORDS PAGEFORMATTINGSTATE PAGEREGION)
    (MACROS GETPFS SETPFS))
  (MACROS \FIRST-COLUMN-START)
  ;; Replaces CL:MULTIPLE-VALUE-SETQ, to avoid CL:VALUES
  (MACROS TEDIT.SETQS TEDIT.VALUES))
(INITRECORDS PAGEREGION)
[COMS ;; Page-numbering font specification/default.
  ;; (Must come before calls to TEDIT.SINGLE.PAGEFORMAT below.)
  (GLOBALVARS TEDIT.DEFAULT.FOLIO.LOOKS)
  (INITVARS (TEDIT.DEFAULT.FOLIO.LOOKS (CHARLOOKS.FROM.FONT (FONTCOPY NIL
    '(FAMILY MODERN SIZE 10 WEIGHT
      MEDIUM SLOPE REGULAR)

[VAR (MAXPAGE# 65535)
  (MINPAGE# 1)
  (TEDIT.PAGE.FRAMES (LIST (TEDIT.SINGLE.PAGEFORMAT NIL NIL NIL NIL NIL 72 72 72 72 NIL 1)
    (TEDIT.SINGLE.PAGEFORMAT T 72 756 NIL 'LEFT 72 72 72 72 NIL 1)
    (TEDIT.SINGLE.PAGEFORMAT T 540 756 NIL 'RIGHT 72 72 72 72 NIL 1]

[COMS ;; Creation, GET, and PUT of page frames.
  (FNS \TEDIT.PARSE.PAGEFRAMES \TEDIT.PUT.PAGEFRAMES \TEDIT.UNPARSE.PAGEFRAMES))
[COMS ;; Public functions for setting up page layouts
  (FNS TEDIT.SINGLE.PAGEFORMAT TEDIT.COMPOUND.PAGEFORMAT TEDIT.PAGEFORMAT))
(FNS TEDIT.FORMAT.HARDCOPY TEDIT.SKIP.SPECIALCOND)
[COMS ;; Perform page layout, based on a regular expression of typed regions.
  (FNS \TEDIT.FORMATBOX \TEDIT.FORMATHEADING \TEDIT.FORMATPAGE \TEDIT.FORMATTEXTBOX
    \TEDIT.FORMATFOLIO \TEDIT.FORMAT.FOUNDBOX?)
  ;; Aux function to capture page headings during line formatting:
  (FNS TEDIT.HARDCOPY.PAGEHEADINGS)
  ;; Aux function to handle end-of-column processing (paragraph keep, widow elimination, etc):
  (FNS TEDIT.HARDCOPY-COLUMN-END))
[COMS ;; Handle varying paper sizes
  (FNS SCALEPAGEUNITS SCALEPAGEXUNITS SCALEPAGEYUNITS \TEDIT.PAPERHEIGHT \TEDIT.PAPERWIDTH)
  (GLOBALVARS TEDIT.PAPER.SIZES)
  (VARS (TEDIT.PAPER.SIZES '( (A0 2384 3370)
    (A1 1684 2384)
    (A2 1191 1684)
    (A3 842 1191)
    (A4 595 842)
    (A5 420 595)
    (B0 2835 4008)
    (B1 2004 2835)
    (B2 1417 2004)
    (B3 1001 1417)
    (B4 709 1001)
    (B5 499 709)

[COMS ; Page numbering option support
  (FNS ROMANNUMERALS))
[COMS ;; Foot note support
  (FNS \TEDIT.FORMAT.FOOTNOTE))))

(DECLARE%: EVAL@COMPILE DONTCOPY
(DECLARE%: EVAL@COMPILE
(RECORD TEDITPAPERSIZE ( ;; Describe the size of a sheet of paper (in points), given a paper size-name.
  TPSNAME ; The name, as a litatom
  TPSWIDTH ; Paper width, in points
  TPSHEIGHT ; Paper Height, in points
  TPSLANDSCAPE? ; T if we have to rotate things to print them on this paper.

)
)
)
```


)

(PUTPROPS TEDIT.VALUES MACRO [ARGS `(LIST ,@ARGS)]
)
)

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

;; Page-numbering font specification/default.
;; (Must come before calls to TEDIT.SINGLE.PAGEFORMAT below.)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.DEFAULT.FOLIO.LOOKS)
)

(RPAQ? TEDIT.DEFAULT.FOLIO.LOOKS [CHARLOOKS.FROM.FONT (FONTCOPY NIL
' (FAMILY MODERN SIZE 10 WEIGHT MEDIUM SLOPE REGULAR
])

(RPAQQ MAXPAGE# 65535)

(RPAQQ MINPAGE# 1)

(RPAQ TEDIT.PAGE.FRAMES
(LIST (TEDIT.SINGLE.PAGEFORMAT NIL NIL NIL NIL NIL 72 72 72 72 NIL 1)
(TEDIT.SINGLE.PAGEFORMAT T 72 756 NIL 'LEFT 72 72 72 72 NIL 1)
(TEDIT.SINGLE.PAGEFORMAT T 540 756 NIL 'RIGHT 72 72 72 72 NIL 1)))

;; Creation, GET, and PUT of page frames.

(DEFINEQ

(\TEDIT.PARSE.PAGEFRAMES

[LAMBDA (PAGELIST PARENT)

; Edited 13-Nov-2023 00:14 by rmk
; Edited 4-Oct-2022 16:57 by rmk
(* jds "31-Jul-84 15:30")

;; Internalize an external pageframe.
;; Exactly like TEDIT.PARSE.PAGEFRAMES1, except this doesn't scale the region specs

(LET (FRAMETYPE PAGEFRAME)
(COND
((type? PAGEREGION PAGELIST)
PAGELIST)
((NEQ 'LIST (SETQ FRAMETYPE (pop PAGELIST)))
[SETQ PAGEFRAME (create PAGEREGION
REGIONFILLMETHOD _ FRAMETYPE
REGIONTYPE _ (pop PAGELIST)
REGIONLOCALINFO _ (pop PAGELIST)
REGIONSPEC _ (OR (pop PAGELIST)
(LIST 0 0 0 0])
(replace REGIONSUBBOXES of PAGEFRAME with (for ALIST in (pop PAGELIST) collect (\TEDIT.PARSE.PAGEFRAMES
ALIST PAGEFRAME))
PAGEFRAME)
(T (for FRAMESPEC in (CAR PAGELIST) collect (\TEDIT.PARSE.PAGEFRAMES FRAMESPEC NIL])

(\TEDIT.PUT.PAGEFRAMES

[LAMBDA (LOOKSFILE PAGEFRAMES)

; Edited 22-Dec-2023 09:03 by rmk
; Edited 19-Dec-2023 10:25 by rmk
; Edited 26-Aug-2023 08:29 by rmk
(* jds "13-Nov-86 20:10")

;; Put out a description of a set of page-layout frames
(\DWOUT LOOKSFILE 0) ; The length of this run of looks
(\WOUT LOOKSFILE \PieceDescriptorPAGEFRAME) ; Mark this as a set of page frames
(PRIN4 (\TEDIT.UNPARSE.PAGEFRAMES PAGEFRAMES)
LOOKSFILE *TEDIT-FILE-READTABLE*])

(\TEDIT.UNPARSE.PAGEFRAMES

[LAMBDA (PAGEFRAME)

; Edited 22-Dec-2023 09:04 by rmk
(* jds "31-Jul-84 15:00")

;; Take an internal page frame, and create an equivalent list structure.
(COND
[(LISTP PAGEFRAME)
(LIST 'LIST (for FRAME in PAGEFRAME collect (\TEDIT.UNPARSE.PAGEFRAMES FRAME])
(T (LIST (fetch REGIONFILLMETHOD of PAGEFRAME)
(fetch REGIONTYPE of PAGEFRAME)
(fetch REGIONLOCALINFO of PAGEFRAME)
(fetch REGIONSPEC of PAGEFRAME)

```

)
(for SUBREGION in (fetch REGIONSUBBOXES of PAGEFRAME) collect (\TEDIT.UNPARSE.PAGEFRAMES SUBREGION]
)
)

```

:: Public functions for setting up page layouts

(DEFINEQ

(TEDIT.SINGLE.PAGEFORMAT

```

[LAMBDA (PAGE#S? PX PY PFONT PQUAD LEFT RIGHT TOP BOTTOM COLS COLWIDTH INTERCOL HEADINGS UNITS PAGEPROPS
PAPERSIZE)
; Edited 13-Nov-2023 08:59 by rmk
; Edited 10-Aug-2023 08:14 by rmk
; Edited 17-Dec-87 14:54 by jds

```

:: Given a description in the args, create a pageframe to describe a single kind of page.

```

(LET* ((LANDSCAPE? (LISTGET PAGEPROPS 'LANDSCAPE?))
(PAPERWIDTH (\TEDIT.PAPERWIDTH PAPERSIZE LANDSCAPE?))
(PAPERHEIGHT (\TEDIT.PAPERHEIGHT PAPERSIZE LANDSCAPE?))
[PAGEREGION (create PAGEREGION
REGIONFILLMETHOD _ 'PAGE
REGIONSPEC _
(create REGION
LEFT _ 0
BOTTOM _ 0
WIDTH _ PAPERWIDTH
HEIGHT _ PAPERHEIGHT)
REGIONLOCALINFO _ (CONS 'PAPERSIZE (CONS PAPERSIZE PAGEPROPS])
PAGEWIDTH SUBREGIONS FOLIOLEFT SCALEFACTOR HEADINGREGIONS)
(SELECTQ UNITS
((POINTS NIL) ; If units are in printers points, the default, do no scaling
(SETQ SCALEFACTOR 1))
(PICAS ; The units are in picas--12pts per. Scale all values.
(SETQ SCALEFACTOR 12))
(INCHES ; The units are in inches, at 72 pts per. Set the scale factor
(SETQ SCALEFACTOR 72))
(MICAS ; The units are MICAS, at 2540 to the inch.
(SETQ SCALEFACTOR 0.02834646))
(CM ; Units are in CM, at 72.27/2.54pts per.
(SETQ SCALEFACTOR (CONSTANT (FQUOTIENT 72 2.54))))
(\ILLEGAL.ARG UNITS) ; We need to do the scaling.
(SETQ PX (SCALEPAGEXUNITS PX SCALEFACTOR PAPERSIZE LANDSCAPE?))
(SETQ PY (SCALEPAGEYUNITS PY SCALEFACTOR PAPERSIZE LANDSCAPE?))
(AND LEFT (SETQ LEFT (HCSCALE SCALEFACTOR LEFT)))
(AND RIGHT (SETQ RIGHT (HCSCALE SCALEFACTOR RIGHT)))
(AND TOP (SETQ TOP (HCSCALE SCALEFACTOR TOP)))
(AND BOTTOM (SETQ BOTTOM (HCSCALE SCALEFACTOR BOTTOM)))
(AND COLWIDTH (SETQ COLWIDTH (HCSCALE SCALEFACTOR COLWIDTH)))
(AND INTERCOL (SETQ INTERCOL (HCSCALE SCALEFACTOR INTERCOL)))
(SETQ PAGEWIDTH (IDIFFERENCE (IDIFFERENCE PAPERWIDTH RIGHT)
LEFT))
(CL:WHEN PAGE#S?
(SELECTQ PQUAD
(LEFT ; If the page number is flush left, set up the region to start where
; he specified.
(SETQ FOLIOLEFT PX))
(RIGHT ; If it's flush right, set up the region to END there
(SETQ FOLIOLEFT (IDIFFERENCE PX 288)))
((CENTERED NIL) ; Otherwise, center the page number around the point he
; specifies
(SETQ FOLIOLEFT (IDIFFERENCE PX 144)))
(SHOULDNT))
:: Note that the folio charlooks is a charlooks spec-list, not a CHARLOOKS. The parse/unparse is just to get the priority union of
:: PFONT with the defaults.
[SETQ SUBREGIONS
(LIST (create PAGEREGION
REGIONFILLMETHOD _ 'FOLIO
REGIONSPEC _ (create REGION
LEFT _ FOLIOLEFT
BOTTOM _ PY
WIDTH _ 288
HEIGHT _ 36)
REGIONLOCALINFO _ `(PARALOOKS [QUAD ,(OR PQUAD 'CENTERED)
CHARLOOKS
, (\TEDIT.UNPARSE.CHARLOOKS.LIST (
\TEDIT.PARSE.CHARLOOKS.LIST
PFONT
TEDIT.DEFAULT.FOLIO.LOOKS
))
FORMATINFO
,(LISTGET PAGEPROPS 'FOLIOINFO])
(CL:WHEN HEADINGS
[SETQ HEADINGREGIONS (for HDG LEFT in HEADINGS
collect

```

:: Run thru the list of headings, building a box for each. By default, a heading will have the same width right margin as the left margin that was specified.


```

    then (CL:UNLESS (\GETSTREAM TEXTSTREAM 'INPUT T)
          [RESETSAVE (SETQ TEXTSTREAM (OPENTEXTSTREAM TEXTSTREAM))
                    `(PROGN (CLOSEF? OLDVALUE))
                    TEXTSTREAM
                    else (ERROR TEXTSTREAM "is not a Tedit stream"))
    (PROG ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TEXTSTREAM))
          [FORMATTINGSTATE (create PAGEFORMATTINGSTATE
                                  PAGE# _ (FIXP FIRSTPG#)
                                  FIRSTPAGE _ T
                                  STATE _ FIRSTPG#
                                  MINPAGE# _ STARTPG
                                  MAXPAGE# _ (OR ENDPG 65535)
                                  CHNO _ 1
                                  PAGEHEADINGS _ (LIST NIL NIL)
                                  PAGE#GENERATOR _ (AND (LISTP FIRSTPG#)
                                                         (CDR FIRSTPG#))
                                  PAGE#TEXT _ (AND (LISTP FIRSTPG#)
                                                    (CAR FIRSTPG#))
                                  PRSTREAM PAGEFRAMES SCRATCHFILE NPAGES WASOPEN TARGETFILENAME)
          (CL:WHEN (EQ 'DON'T (APPLY* (OR (GETTEXTPROP TEXTOBJ 'BEFOREHARDCOPYFN)
                                         (FUNCTION NIL))
                                     TEXTSTREAM))
                  ; Do pre-hardcopy processing as indicated, or refuse
                  (RETURN))
          (SETQ PAGEFRAMES (OR (FGETTOBJ TEXTOBJ TXTPAGEFRAMES)
                              TEDIT.PAGE.FRAMES))
          (CL:WHEN (LISTP PAGEFRAMES)
                  ; If it's a list, pack it into a real set of specs.
                  (SETQ PAGEFRAMES (TEDIT.COMPOUND.PAGEFORMAT (CAR PAGEFRAMES)
                                                                (CADR PAGEFRAMES)
                                                                (CADDR PAGEFRAMES))))
          (SETPFS FORMATTINGSTATE PRESSREGION TEDIT.DEFAULTPAGEREGION)
                  ; Print in the usual region on the page
          (CL:UNLESS BREAKPAGETITLE
                  [SETQ BREAKPAGETITLE (COND
                                          ((LISTGET PRINTOPTIONS 'DOCUMENT.NAME))
                                          ((OR (NOT (FGETTOBJ TEXTOBJ TXTFILE))
                                              (STRINGP (FGETTOBJ TEXTOBJ TXTFILE))
                                              (type? STREAM (fetch (STREAM FULLNAME) of (FGETTOBJ TEXTOBJ TXTFILE)
                                                                    )))
                                              (STRINGP (fetch (STREAM FULLNAME) of (FGETTOBJ TEXTOBJ TXTFILE)
                                                                    ))
                                              ; This isn't a real file, so print a generic name on the document
                                              ; break page.
                                              "Tedit Hardcopy Output")
                                          (T
                                           ; It's a real file, so use the file name on the break page.
                                           (fetch (STREAM FULLNAME) of (FGETTOBJ TEXTOBJ TXTFILE)]
                                          [SETQ SCRATCHFILE (OR FILE (PRINTER.SCRATCH.FILE (TEXTSTREAM TEXTSTREAM))
                                                              ; Set up to do the user's cleanup on the way out, as well.
                                                              (RESETLST
                                                                (TEDIT.PROMPTPRINT TEXTOBJ "Formatting for print..." T)
                                                                [COND
                                                                  ((AND FILE (OPENP FILE)
                                                                    (IMAGESTREAMTYPE FILE))
                                                                    ; The file he handed us is already an image-type file. Just
                                                                    ; append the new stuff to it.
                                                                    (SETQ WASOPEN T)
                                                                    (SETQ PRSTREAM FILE))
                                                                  (T
                                                                    ; T wasn't an image stream, so let's open us one.
                                                                    (RESETSAVE (SETQ PRSTREAM (OPENIMAGESTREAM SCRATCHFILE
                                                                                              [OR IMAGETYPE (SETQ IMAGETYPE
                                                                                              (CAR (PRINTERPROP (PRINTERTYPE SERVER)
                                                                                              'CANPRINT]
                                                                                              (LIST 'FONT (FONTCREATE 'TERMINAL 10)
                                                                                              'BREAKPAGEFILENAME BREAKPAGETITLE)))
                                                                                              ' (AND RESETSTATE (DELFILE (CLOSEF? OLDVALUE))
                                                                                              ; So we close and delete the file in case of trouble.
                                                                                              ;; The right margin must be big enough to prevent line wrap on landscaped 14 inch paper, with Postscript's scaling of .01-point
                                                                                              ;; increments. (~ 101,000). This will cause a performance hit. Sigh. JDS 9/5/89
                                                                                              (DSPRIGHTMARGIN 131072 PRSTREAM)
                                                                                              [while (ILEQ (GETPFS FORMATTINGSTATE CHNO)
                                                                                              (FGETTOBJ TEXTOBJ TEXTLEN))
                                                                                              do ;; Format pages according to the existing layout:
                                                                                              (for PAGEREGION inside PAGEFRAMES do (\TEDIT.FORMATBOX TEXTOBJ PRSTREAM (GETPFS
                                                                                              FORMATTINGSTATE
                                                                                              CHNO)
                                                                                              PAGEREGION FORMATTINGSTATE IMAGETYPE))
                                                                                              (CL:WHEN (EQ (GETPFS FORMATTINGSTATE STATE)
                                                                                              :NEW-PAGE-LAYOUT)
                                                                                              ;; New page layout got specified. Prepare to re-enter the formatting code and skip to the equivalent page in the
                                                                                              ;; new format.
                                                                                              (SETQ PAGEFRAMES (GETPFS FORMATTINGSTATE NEWPAGE_LAYOUT))
                                                                                              ;; Set up the formatting state so code knows we're looking for an equivalent page, and which page it is. (The SUB1
                                                                                              ;; is because we counted an extra page for the page on which the new payout was detected.)
                                                                                              (SETPFS FORMATTINGSTATE REQUIREDREGIONTYPE (SUB1 (GETPFS FORMATTINGSTATE PAGECOUNT))
                                                                                              )
                                                                                              (SETPFS FORMATTINGSTATE PAGECOUNT 0)
                                                                                              (SETPFS FORMATTINGSTATE STATE :SEARCHING-FOR-EQUIVALENT-PAGE)

```

```

(CL:WHEN (LISTP PAGEFRAMES) ; If it's a list, pack it into a real set of specs.
  (SETQ PAGEFRAMES (TEDIT.COMPOUND.PAGEFORMAT (CAR PAGEFRAMES)
    (CADR PAGEFRAMES)
    (CADDR PAGEFRAMES))))))
(SETQ TARGETFILENAME (STREAMPROP PRSTREAM 'PDFTARGETINFO))
(CL:UNLESS WASOPEN ; Only if we created the image stream should we close it.
  (SETQ PRSTREAM (CLOSEF PRSTREAM))
  (CL:UNLESS DONTSEND
    (SEND.FILE.TO.PRINTER PRSTREAM SERVER (APPEND PRINTOPTIONS (LIST 'DOCUMENT.NAME
      BREAKPAGETITLE))))))
(CL:UNLESS FILE (DELFILE SCRATCHFILE))
(APPLY* (OR (GETTEXTPROP TEXTOBJ 'AFTERHARDCOPYFN)
  (FUNCTION NIL))
  TEXTSTREAM)
(SETQ NPAGES (GETPFS FORMATTINGSTATE PAGECOUNT))
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT NPAGES " page" (CL:IF (EQ 1 NPAGES)
  ""
  "s")
  " printed"
  (CL:IF (EQ FILE SCRATCHFILE)
    (CONCAT " to " (OR TARGETFILENAME (FULLNAME FILE)))
    ""))
  T)
(RETURN NPAGES)))

```

(TEDIT.SKIP.SPECIALCOND

```
[LAMBDA (TSTREAM LINE PARALOOKS CHNO)
```

```

; Edited 5-Jul-2023 14:19 by rmk
; Edited 15-May-2023 22:36 by rmk
; Edited 16-Feb-2023 00:08 by rmk
; Edited 25-May-93 13:44 by sybalsky:mv:envos

```

```

;; This is a special-paragraph that should be skipped in this context (e.g. an EVEN para on an odd page). This is done by setting LINE:LCHARLIM
;; to the last character of the heading so it will move the document ahead to the next real text.

```

```

(SETLD LINE LMARK 'SPECIAL)
(SETLD LINE 1STLN T)
(SETLD LINE LSTLN T)
(SETLD LINE LHEIGHT 0)
(SETLD LINE ASCENT 0)
(SETLD LINE DESCENT 0)
(SETLD LINE LTRUEASCENT 0)
(SETLD LINE LTRUEDESCENT 0)
(SETLD LINE LCHARLIM (SUB1 (IPLUS CHNO (for PC (HEADINGTYPE _ (fetch (FMTSPEC FMTPARASUBTYPE) of PARALOOKS))
  inpieces
  (fetch (TEXTSTREAM PIECE) of TSTREAM)
  while [AND (EQ 'PAGEHEADING (fetch (FMTSPEC FMTPARATYPE)
    of (PPARALOOKS PC)))
    (EQ HEADINGTYPE (fetch (FMTSPEC FMTPARASUBTYPE)
    of (PPARALOOKS PC]
  sum (PLEN PC]

```

)

```
;; Perform page layout, based on a regular expression of typed regions.
```

```
(DEFINEQ
```

(\TEDIT.FORMATBOX

```
[LAMBDA (TEXTOBJ PRSTREAM STARTINGCHNO PAGEREGION FORMATTINGSTATE)
```

```

; Edited 13-Mar-2024 17:09 by rmk
; Edited 20-Jan-2024 12:16 by rmk
; Edited 28-Jun-2023 15:54 by rmk
; Edited 22-Jun-2023 21:50 by rmk
; Edited 9-May-2023 18:22 by rmk
; Edited 15-Feb-2023 23:47 by rmk
; Edited 30-May-91 12:51 by jds

```

```

;; Grab text from the TEXTOBJ, starting with STARTINGCHNO, and use it to fill REGION on a page. Return a list of line descriptors which, taken
;; together, fill the region.

```

```
;; This updates the CHNO field of the PAGEFORMATTINGSTATE
```

```

(LET ((REGION (fetch (PAGEREGION REGIONSPEC) of PAGEREGION))
  CHNO LINES LAST-CHNO SUBREGIONSPEC (TEXTLEN (FGETTOBJ TEXTOBJ TEXTLEN)))
  (SELECTQ (fetch REGIONFILLMETHOD of PAGEREGION)
    (TEXT ; A normal text region. Fill it with text formatted the usual way.
      (CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
        ; Only format if we're not looking for something else.
        (TEDIT.SETQS (LINES NIL LAST-CHNO)
          (\TEDIT.FORMATTEXTBOX TEXTOBJ PRSTREAM STARTINGCHNO PAGEREGION FORMATTINGSTATE))))
    (FOLIO ; A Page Number. Fill it in according to the instructions
      (CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
        ; Only format if we're not looking for something else.
        (SETQ LINES (\TEDIT.FORMATFOLIO TEXTOBJ PRSTREAM FORMATTINGSTATE PAGEREGION))))
    (HEADING ; A Page heading. Fill it in from a text source we saved for the
      ; occasion.
      (CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)

```

)


```

; Only format if we're not looking for something else.
      (SETQ LINES (\TEDIT.FORMATHEADING TEXTOBJ PRSTREAM FORMATTINGSTATE PAGEREGION)))
(PAGE ;; This box is really a PAGE FRAME, no lines here. Fill it in and do whatever other processing is needful for end of page.
  (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
      ; So that if this is the box he's looking for, we'll spot it and stop
      ; searching
  (\TEDIT.FORMATPAGE TEXTOBJ PRSTREAM STARTINGCHNO PAGEREGION FORMATTINGSTATE))
((RECURSIVE SEQUENCE ALTERNATE SELECTION REPEAT) ; This box is really a list of boxes. Fill them.
  (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
      ; So that if this is the box he's looking for, we'll spot it and stop
      ; searching
  (SELECTQ (fetch REGIONFILLMETHOD of PAGEREGION)
    ((SEQUENCE RECURSIVE) ; Just run thru filling in the sub-boxes in order.
      (bind SUBREGIONSPEC for SUBREGION in (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION)
        while (AND (ILEQ (GETPFS FORMATTINGSTATE CHNO)
          TEXTLEN)
            (OR (NOT (GETPFS FORMATTINGSTATE PAGE#))
              (NOT (GETPFS FORMATTINGSTATE MAXPAGE#))
              (ILEQ (GETPFS FORMATTINGSTATE PAGE#)
                (GETPFS FORMATTINGSTATE MAXPAGE#)))
            (NEQ (GETPFS FORMATTINGSTATE STATE)
              :NEW-PAGE-LAYOUT))
          do [SETQ SUBREGIONSPEC (create REGION using (fetch REGIONSPEC of SUBREGION)
            LEFT _
              (IPLUS (fetch (REGION LEFT)
                of (fetch REGIONSPEC
                  of SUBREGION))
                (fetch (REGION LEFT) of REGION))
            BOTTOM _
              (IPLUS (fetch (REGION BOTTOM)
                of (fetch REGIONSPEC
                  of SUBREGION))
                (fetch (REGION BOTTOM)
                  of REGION))
            (\TEDIT.FORMATBOX TEXTOBJ PRSTREAM (GETPFS FORMATTINGSTATE CHNO)
              (create PAGEREGION using SUBREGION REGIONSPEC _ SUBREGIONSPEC)
              FORMATTINGSTATE)))
          (ALTERNATE ; Run through the sub-boxes repeatedly in sequence.
            (while (AND (ILEQ (GETPFS FORMATTINGSTATE CHNO)
              TEXTLEN)
                (NEQ :NEW-PAGE-LAYOUT (GETPFS FORMATTINGSTATE STATE)))
              do (bind SUBREGIONSPEC for SUBREGION in (fetch (PAGEREGION REGIONSUBBOXES)
                of PAGEREGION)
                while (AND (ILEQ (GETPFS FORMATTINGSTATE CHNO)
                  TEXTLEN)
                    (NEQ (GETPFS FORMATTINGSTATE STATE)
                      :NEW-PAGE-LAYOUT))
                  do [SETQ SUBREGIONSPEC (create REGION
                    using (fetch REGIONSPEC of SUBREGION)
                    LEFT _
                      (IPLUS (fetch (REGION LEFT)
                        of (fetch REGIONSPEC
                          of SUBREGION))
                        (fetch (REGION LEFT)
                          of REGION))
                    BOTTOM _
                      (IPLUS (fetch (REGION BOTTOM)
                        of (fetch REGIONSPEC
                          of SUBREGION))
                        (fetch (REGION BOTTOM)
                          of REGION))
                    (\TEDIT.FORMATBOX TEXTOBJ PRSTREAM (GETPFS FORMATTINGSTATE CHNO)
                      (create PAGEREGION using SUBREGION REGIONSPEC _ SUBREGIONSPEC
                        )
                      FORMATTINGSTATE)))
                (SELECTION) ; Do one or another box, depending on some criterion.
                (SHOULDNT)) ; For now, draw a box around it, too.
              )
            NIL)
          (for LINE LTEXTOBJ in LINES when LINE do ; Run thru the lines displaying them all.
            (BLOCK)
            (SETQ LTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ)
              of (FGETLD LINE LTEXTSTREAM)))
            (CL:WHEN (OR (NOT (GETPFS FORMATTINGSTATE MINPAGE#))
              (IGEQ (GETPFS FORMATTINGSTATE PAGE#)
                (GETPFS FORMATTINGSTATE MINPAGE#)))
              ; We're beyond the min page number -- go ahead and print the
              ; line
              (\TEDIT.HARDCOPY.DISPLAYLINE LTEXTOBJ LINE
                (SCALEREGION (DSPSCALE NIL PRSTREAM)
                  REGION)
                PRSTREAM FORMATTINGSTATE))
              (CL:WHEN (EQ TEXTOBJ LTEXTOBJ)
                ;; This line refers back to the main text, so update the current-char pointer.

```

:: [NB that footnotes could cause the count to be non-monotonic; hence the
:: IMAX.]

[SETQ CHNO (IMAX (OR CHNO 0)
(ADD1 (FGETLD LINE LCHARLIM))

(push (GETPFS FORMATTINGSTATE PAGELINECACHE)
LINE)
(FSETLD LINE LTEXTSTREAM NIL))

(COND
(LAST-CHNO ; We got a definite last chno from FORMATTEXTBOX.
(SETPFS FORMATTINGSTATE CHNO LAST-CHNO))
(CHNO ; Otherwise, use the new char no if we computed one.
(SETPFS FORMATTINGSTATE CHNO CHNO))

(\TEDIT.FORMATHEADING

[LAMBDA (TEXTOBJ PRSTREAM FORMATTINGSTATE PAGEREGION) ; Edited 17-Mar-2024 00:24 by rmk
; Edited 15-Mar-2024 19:23 by rmk
; Edited 13-Mar-2024 09:00 by rmk
; Edited 6-Mar-2024 13:09 by rmk
; Edited 15-Feb-2024 22:02 by rmk
; Edited 19-Jan-2024 23:20 by rmk
; Edited 9-Sep-2023 22:17 by rmk
; Edited 19-May-2023 21:15 by rmk
; Edited 9-May-2023 20:30 by rmk
; Edited 9-Oct-90 13:24 by jds

:: Grab heading pieces from the FORMATTINGSTATE and use them to fill REGION on a page. Return a list of line descriptors which fill the region.

(LET ((REGION (SCALEREGION (DSPSCALE NIL PRSTREAM)
(fetch (PAGEREGION REGIONSPEC) of PAGEREGION)))
(LOCALINFO (fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION))
HEADINGTEXTOBJ HEADINGSTREAM FORCENEXTPAGE HEADING)
(CL:WHEN [AND (for FORM inside (LISTGET LOCALINFO 'PRECONDITIONS) always (EVAL FORM))
(SETQ HEADING (LISTGET (GETPFS FORMATTINGSTATE PAGEHEADINGS)
(LISTGET LOCALINFO 'HEADINGTYPE]

:: Bind the stream to make sure it isn't collected.

[SETQ HEADINGSTREAM (OPENTEXTSTREAM NIL NIL NIL NIL '(PARALOOKS , (PPARALOOKS (fetch (SELPIECES
SPFIRST)
of HEADING]

(SETQ HEADINGTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of HEADINGSTREAM))

:: Insert the heading pieces into HEADINGTEXTOBJ

(\TEDIT.INSERTPIECES (fetch (SELPIECES SPFIRST) of HEADING)
(\TEDIT.ALIGNEDPIECE 1 HEADINGTEXTOBJ)
HEADINGTEXTOBJ)

::

:: Why is BOTTOM said to be the "top" of the region to be filled?

(bind LINE YBOT (BOTTOM _ (fetch (REGION BOTTOM) of REGION))
(LEN _ (TEXTLEN HEADINGTEXTOBJ))
(CHNO _ 1) while (ILESSP CHNO LEN) until FORCENEXTPAGE
collect

:: Format the next line from HEADINGTEXTOBJ pieces

(SETQ LINE (\TEDIT.FORMATLINE HEADINGTEXTOBJ CHNO NIL REGION PRSTREAM FORMATTINGSTATE))
(SETQ FORCENEXTPAGE (EQ (CHARCODE FORM)
(GETLD LINE FORCED-END)))

[SETQ YBOT (COND
(YBOT ; Take account of this line's height
(IDIFFERENCE YBOT (FGETLD LINE LHEIGHT)))
(T ; First line: position it at the top of the region.
(IDIFFERENCE BOTTOM (FGETLD LINE DESCENT)]
(SETYPOS LINE YBOT)
(SETQ CHNO (ADD1 (FGETLD LINE LCHARLIM))) ; Set the start of the next line
LINE))]

(\TEDIT.FORMATPAGE

[LAMBDA (TEXTOBJ PRSTREAM CHNO PAGEREGION FORMATTINGSTATE) ; Edited 17-Mar-2024 00:24 by rmk
; Edited 13-Mar-2024 10:28 by rmk
; Edited 19-Jan-2024 23:10 by rmk
; Edited 11-Dec-2023 22:02 by rmk
; Edited 13-Nov-2023 00:15 by rmk
; Edited 22-Sep-2023 20:37 by rmk
; Edited 15-Jul-2023 22:22 by rmk
; Edited 5-Jul-2023 12:49 by rmk
; Edited 8-Mar-2023 18:20 by rmk
; Edited 4-Mar-2023 22:10 by rmk
; Edited 9-Oct-2022 17:24 by rmk
; Edited 4-Jul-93 00:29 by sybalsky:MV:ENVOS

:: Format a whole page -- run thru the page's sub-boxes filling them in by type:

- :: FOLIO -- page number
- :: PAGEHEADING -- running heads/footers
- :: TEXT -- plain running text.

```
(CL:UNLESS (EQ :SEARCHING-FOR-EQUIVALENT-PAGE (GETPFS FORMATTINGSTATE STATE))
;; Only do real page formatting work if we're not trying to get ourselves to an equivalent page frame spec (having switched page layouts in
;; mid-document).
[PROG ((PAGE# (GETPFS FORMATTINGSTATE PAGE#))
(PAGEPROPS (fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION))
(REGION (\TEDIT.SCALEREGION (DSPSCALE NIL PRSTREAM)
(fetch (PAGEREGION REGIONSPEC) of PAGEREGION)))
(END-OF-PAGE-FN (GETTEXTPROP TEXTOBJ 'END-OF-PAGE-FN))
(PRE-EXISTING-FONT (DSPFONT NIL PRSTREAM))
(TEXTLEN (TEXTLEN TEXTOBJ))
END-OF-PAGE-MARKER STARTING-FILEPTR PC NEWPARALOOKS)
;; For real page independence, we need to reset the font to where it was as of the beginning of the page before calling DSPNEWPAGE.
;; This avoids font creation in a page prolog, which might get missed otherwise.
; Print in the usual region on the page
(CL:UNLESS (ILEQ CHNO TEXTLEN)
(RETURN))
(SETQ PC (\TEDIT.ALIGNEDPIECE CHNO TEXTOBJ))
(SETQ NEWPARALOOKS (\TEDIT.APPLY.PARASTYLES (PPARALOOKS PC)
PC TEXTOBJ)) ;RMK: Why both 'NEWPAGE LAYOUT and
;:NEW-PAGE-LAYOUT ?
(CL:WHEN (EQ 'NEWPAGE LAYOUT (fetch (FMTSPEC FMTPARATYPE) of NEWPARALOOKS))
;; The first paragraph on this page starts a new page layout.
(SETPFS FORMATTINGSTATE STATE :NEW-PAGE-LAYOUT)
;; The first character of the paragraph after the one containing PC:
[SETPFS FORMATTINGSTATE CHNO (ADD1 (CAR (\TEDIT.PARA.LAST TEXTOBJ PC))
[SETPFS FORMATTINGSTATE NEWPAGE LAYOUT (\TEDIT.PARSE.PAGEFRAMES (LISTGET (fetch (FMTSPEC
FMTUSERINFO
)
of NEWPARALOOKS)
'NEWPAGE LAYOUT]
(RETURN))
;;
(CL:UNLESS PAGE#
;; If this page template specifies a starting page number, use it.
(SETQ PAGE# (OR (LISTGET PAGEPROPS 'STARTINGPAGE#)
1))
(SETPFS FORMATTINGSTATE PAGE# PAGE#))
(CL:WHEN (LISTGET PAGEPROPS 'LANDSCAPE?) ; This is a landscape page.
(STREAMPROP PRSTREAM 'PRINTERMODE 'LANDSCAPE) ; Put the info. into stream
(DSPPUSHSTATE PRSTREAM)
(DSPROTATE 90 PRSTREAM)
(DSPTRANSLATE 0 (IMINUS (ffetch (REGION HEIGHT) of REGION))
PRSTREAM))
(DSPCLIPPINGREGION REGION PRSTREAM) ; Clip to the whole sheet.
(DSPRIGHTMARGIN (ffetch (REGION WIDTH) of REGION)
PRSTREAM)
;; Go thru any leading page heading paras on the page, collecting copies of those pieces in the FORMATTINGSTATE. The value is the first
;; CHNO of the start of the first non-heading piece.
(SETQ CHNO (TEDIT.HARDCOPY.PAGEHEADINGS TEXTOBJ CHNO FORMATTINGSTATE))
;;
;; We now fill up the next complete page. Afterwards, we either continue to the next page (DPSNEWPAGE) or finish up.
;; TEDIT.FORMATBOX is responsible for setting up NEWPAGEBEFORFE and NEWPAGEAFTER
(SETPFS FORMATTINGSTATE CHNO CHNO)
(for SUBREGION in (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION)
while (ILEQ (GETPFS FORMATTINGSTATE CHNO)
TEXTLEN)
do ;; Now format the subregions of the page. The CHNO field may be updated by each call.
(\TEDIT.FORMATBOX TEXTOBJ PRSTREAM (GETPFS FORMATTINGSTATE CHNO)
SUBREGION FORMATTINGSTATE))
;;
(DSPFONT PRE-EXISTING-FONT PRSTREAM)
(CL:WHEN (LISTGET PAGEPROPS 'LANDSCAPE?) ; This is a landscape page.
(STREAMPROP PRSTREAM 'PRINTERMODE NIL)
(DSPTRANSLATE 0 (ffetch (REGION HEIGHT) of REGION)
PRSTREAM)
(DSPROTATE 0 PRSTREAM)
(DSPPOPSTATE PRSTREAM))
[COND
([AND (ILEQ (GETPFS FORMATTINGSTATE CHNO)
TEXTLEN)
[NOT (AND END-OF-PAGE-FN (EQ 'DON'T (SETQ END-OF-PAGE-MARKER (APPLY* END-OF-PAGE-FN
TEXTOBJ
FORMATTINGSTATE))
[NOT (AND (GETPFS FORMATTINGSTATE MINPAGE#)
(ILESSP PAGE# (GETPFS FORMATTINGSTATE MINPAGE#))
(NOT (AND (GETPFS FORMATTINGSTATE MAXPAGE#)

```

```

      (IEQ PAGE# (GETPFS FORMATTINGSTATE MAXPAGE#]
      ; There is more to print...
      ; Force the new page
      (DSPNEWPAGE PRSTREAM))
      ((OR (AND (GETPFS FORMATTINGSTATE MAXPAGE#)
                (IGEQ PAGE# (GETPFS FORMATTINGSTATE MAXPAGE#)))
          (EQ END-OF-PAGE-MARKER 'DON'T)))
      ; We've run past the last page to be formatted. or were told to
      ; stop.
      (SETPFS FORMATTINGSTATE CHNO (ADD1 TEXTLEN]
      (add (GETPFS FORMATTINGSTATE PAGE#)
            1)
      (SETPFS FORMATTINGSTATE FIRSTPAGE NIL)
      (SETPFS FORMATTINGSTATE PAGE#TEXT (pop (GETPFS FORMATTINGSTATE PAGE#GENERATOR]))
      ;; Some things happen regardless of whether we're searching or not: Need to count pages we pass over to find an equivalent page in the new
      ;; layout:
      (add (GETPFS FORMATTINGSTATE PAGECOUNT)
            1])

```

(\TEDIT.FORMATTEXTBOX

```

[LAMBDA (TEXTOBJ PRSTREAM CHNO PAGEREGION FORMATTINGSTATE)
; Edited 15-Mar-2024 19:24 by rmk
; Edited 19-Jan-2024 23:37 by rmk
; Edited 4-Dec-2023 12:34 by rmk
; Edited 4-Jul-2023 08:02 by rmk
; Edited 2-Jul-2023 20:49 by rmk
; Edited 1-Jun-2023 15:32 by rmk
; Edited 27-May-2023 12:19 by rmk
; Edited 30-Sep-2022 10:06 by rmk
; Edited 24-Aug-2022 11:45 by rmk
; Edited 3-Jul-93 22:14 by sybalsky:MV:ENVOS

```

;; Grab text from the TEXTOBJ, starting with CH#, and use it to fill REGION on a page. Return a list of line descriptors which, taken together, fill the region.

```

(CL:UNLESS (EQ (GETPFS FORMATTINGSTATE STATE)
               'SEARCHING)

```

;; Only format text if we're really formatting.

```

(LET* ((REGION (SCALEREGION (DSPSCALE NIL PRSTREAM)
                             (ffetch (PAGEREGION REGIONSPEC) of PAGEREGION)))
       (COLUMNBOTTOM (fetch (REGION BOTTOM) of REGION))
       (RTOP (fetch (REGION TOP) of REGION))
       (FIRSTLINE T)
       (FOOTNOTELINES (ffetch PAGEFOOTNOTELINES of FORMATTINGSTATE))
       (FORCENEXTPAGE PAGEFOOTNOTES PRIOR-COLUMN-YBOT LINES ORPHAN FINAL-CHNO FOOTNOTE-REMNANTS)

```

;; Account for lines carried over from prior columns:

```

(bind LINE KEPT-ONE-LINE while (AND (ILEQ COLUMNBOTTOM RTOP)
                                     (SETQ LINE (pop FOOTNOTELINES)))

```

do ;; Move as many potential footnote lines into this column as will fit. And move the bottom of the column up to account for them.

```

(CL:WHEN (IGREATERP (+ COLUMNBOTTOM (FGETLD LINE LHEIGHT))
                   RTOP)

```

;; If we ran out of room for footnotes, put this line back on the queue

```

(TEDIT.SETQS (PAGEFOOTNOTES FOOTNOTE-REMNANTS KEPT-ONE-LINE)
              (TEDIT.HARDCOPY-COLUMN-END PAGEFOOTNOTES LINE NIL 1 NIL REGION TEXTOBJ
              FORMATTINGSTATE))

```

```

(CL:WHEN KEPT-ONE-LINE
  (add COLUMNBOTTOM (FGETLD LINE LHEIGHT)))
(SETQ FOOTNOTELINES (APPEND FOOTNOTE-REMNANTS FOOTNOTELINES))
(RETURN)

```

```

(SETQ PAGEFOOTNOTES (NCONC1 PAGEFOOTNOTES LINE))
(add COLUMNBOTTOM (FGETLD LINE LHEIGHT))

```

```

(SETPFS FORMATTINGSTATE PAGEFOOTNOTELINES FOOTNOTELINES)

```

; Remember any remaining footnotes

```

[SETQ LINES

```

```

(bind LINE FMTSPEC LHEIGHT PREVLIN SPECIALYPOS BREAKAFTERLASTPARA YBOT NEWPAGETYPE COLUMN-YBASE
      (TEXTLEN _ (TEXTLEN TEXTOBJ)) while (AND (ILEQ CHNO TEXTLEN)
                                                (NOT FORCENEXTPAGE))

```

collect (BLOCK)

;; Grab a line descriptor from the recycling list, or create a new one.

```

(SETQ LINE (pop (GETPFS FORMATTINGSTATE PAGELINECACHE)))
; Format the line, noting any form-feeds
(SETQ LINE (\TEDIT.FORMATLINE TEXTOBJ CHNO LINE REGION PRSTREAM FORMATTINGSTATE))
(SETQ FORCENEXTPAGE (AND (EQ (CHARCODE FORM)
                              (FGETLD LINE FORCED-END))
                          'USERBREAK))

```

```

(SETQ LHEIGHT (FGETLD LINE LHEIGHT))
(SETQ FMTSPEC (FGETLD LINE LFMTSPEC))
(COND

```

```

  ((FGETLD LINE LMARK)

```

;; This line is a placeholder for a page heading, OR for a conditional line that is to be skipped (e.g., an EVEN text para on an odd page). All it tells us is what character to skip to so we can continue.

```

(SETQ CHNO (ADD1 (FGETLD LINE LCHARLIM)))
LINE)
((LISTGET (fetch (FMTSPEC FMTUSERINFO) of FMTSPEC)
'FOOTNOTE)
;; This paragraph is a footnote para.
(CL:WHEN FORCENEXTPAGE ; HELP in original code. SHOULDNT ?
(SHOULDNT))
(SETQ FOOTNOTELINES (\TEDIT.FORMAT.FOOTNOTE TEXTOBJ PRSTREAM LINE REGION
FORMATTINGSTATE))
(SETQ CHNO (ADD1 (FGETLD (CAR (FLAST FOOTNOTELINES))
LCHARLIM))) ; Grab the lines of this footnote
[COND
[(GETPFS FORMATTINGSTATE PAGEFOOTNOTELINES)
;; There are overflow footnote lines from this page already. Add to them.
(SETPFS FORMATTINGSTATE PAGEFOOTNOTELINES (COPY (APPEND (GETPFS FORMATTINGSTATE
PAGEFOOTNOTELINES
)
FOOTNOTELINES)]
(T ;; No overflow footnote lines yet. Try adding more footnotes to this page/column.
(for LTAIL LINE on FOOTNOTELINES
do (SETQ LINE (CAR LTAIL))
(add COLUMNBOTTOM LHEIGHT)
(CL:WHEN (IGREATERP COLUMNBOTTOM (OR YBOT RTOP))
;; This one overflows
(TEDIT.SETQS (PAGEFOOTNOTES FOOTNOTE-REMNANTS)
(\TEDIT.HARDCOPY-COLUMN-END PAGEFOOTNOTES LINE NIL 1 NIL
REGION TEXTOBJ FORMATTINGSTATE 3 (NOT FIRSTLINE)))
[SETPFS FORMATTINGSTATE PAGEFOOTNOTELINES (COPY (APPEND
FOOTNOTE-REMNANTS
(CDR LTAIL)
[SETQ FINAL-CHNO (IMAX CHNO (ADD1 (GETLD (CAR (FLAST LTAIL))
LCHARLIM]
(RETURN))
(SETQ PAGEFOOTNOTES (NCONC1 PAGEFOOTNOTES LINE)
;; Don't accumulate footnote lines.
NIL)
(T ;; This line is not a page heading or a footnote, format it.
(SETQ SPECIALYPOS NIL)
;; So that only the first line of a specially-placed paragraph is guaranteed to appear in the current box.
[SETQ YBOT (COND
((AND (fetch (FMTSPEC FMTSPECIALY) of FMTSPEC)
(NOT (ZEROP (fetch (FMTSPEC FMTSPECIALY) of FMTSPEC)))
(FGETLD LINE 1STLN))
; There is a special Y location for this paragraph. Move there
(SETQ SPECIALYPOS (ffetch (FMTSPEC FMTSPECIALY) of FMTSPEC)))
((AND COLUMN-YBASE (FGETLD LINE 1STLN)
(EQ (ffetch (FMTSPEC FMTCOLUMN) OF FMTSPEC)
'NEXT))
;; This is the first line of a new column; back YBOT back down to match the prior column.
(- COLUMN-YBASE (FGETLD LINE DESCENT)))
[YBOT
;; We're into it; take account of this line's height. Original code did the complicated LHEIGHT
;; calculation and threw it away. I assume that that was an error, that the new setting of LHEIGHT is
;; for the benefit of the new YBOT value (which I pulled out of an alternative branch of a COND.
(CL:WHEN (fetch (FMTSPEC FMTBASETOBASE) of FMTSPEC)
[SETQ LHEIGHT (IPLUS (FGETLD LINE DESCENT)
(fetch (FMTSPEC FMTBASETOBASE)
of FMTSPEC)
(COND
[(FGETLD LINE 1STLN)
(IPLUS (fetch (FMTSPEC LEADBEFORE)
of FMTSPEC)
(fetch (FMTSPEC LEADAFTER)
of (GETLD PREVLIN
LFMTSPEC]
(T 0])
(COND
((\FIRST-COLUMN-START LINE FMTSPEC)
(IDIFFERENCE (IMIN PRIOR-COLUMN-YBOT YBOT)
LHEIGHT))
(T (IDIFFERENCE YBOT LHEIGHT)
(T ; Just starting out; find the line's position with respect to the top
; of the region to be filled.
(IDIFFERENCE RTOP (IPLUS (FGETLD LINE LTRUEASCENT)
(FGETLD LINE DESCENT]
(COND
((AND (ILESSP YBOT COLUMNBOTTOM)
(NOT SPECIALYPOS))

```

:: This line hangs off the bottom; (and isn't the first line of a specially-placed paragraph) punt it.

```
(SETQ FORCENEXTPAGE T)
(SETQ FINAL-CHNO (FGETLD LINE LCHAR1))
(SETQ ORPHAN LINE) ; Remember this potential orphan
NIL)
(AND (NOT FIRSTLINE)
      (FGETLD LINE 1STLN)
      (SETQ NEWPAGETYPE (OR (fetch (FMTSPEC FMTNEWPAGEBEFORE)
                                   of (FGETLD LINE LFMTSPEC))
                            BREAKAFTERLASTPARA)))
```

:: We're supposed to put this line at the start of a new page/column (any box, later)

```
(SETQ FORCENEXTPAGE 'USERBREAK)
(SETQ FINAL-CHNO (FGETLD LINE LCHAR1))
(SETQ ORPHAN NIL)
(CL:UNLESS (EQ NEWPAGETYPE T) ; This isn't simply go to a new box; we need to set up the search
           ; for it.
           (SETPFS FORMATTINGSTATE STATE 'SEARCHING)
           (SETPFS FORMATTINGSTATE REQUIREDREGIONTYPE NEWPAGETYPE)))
```

NIL)

(T ; This line is good; use it.

```
(CL:WHEN (AND (fetch (FMTSPEC FMTNEWPAGEAFTER) of FMTSPEC))
          ; We're supposed to put the line after this one at the start of a
          ; new page/column (any box, later)
          (SETQ BREAKAFTERLASTPARA T))
(SETQ PRIOR-COLUMN-YBOT (CL:IF PRIOR-COLUMN-YBOT
                              (IMIN PRIOR-COLUMN-YBOT YBOT)
                              YBOT))
```

```
(SETYPOS LINE YBOT)
(CL:WHEN (\FIRST-COLUMN-START LINE FMTSPEC)
```

:: This is the start of a new group of paragraphs to be lined up in columns. Save the YBASE for
:: these guys for the other columns.

```
(SETQ COLUMN-YBASE (GETLD LINE YBASE)))
(SETQ FIRSTLINE NIL) ; Note that we have put text out on this page/column/box, for first
                    ; line checking.
(SETQ CHNO (ADD1 (GETLD LINE LCHARLIM))) ; Keep track of the next character...
```

```
(SETQ PREVLIN LINE)
LINE]
```

(SETQ LINES (DREMOVE NIL LINES)) ; Remove any NILs from the line list; they're artifacts of running
; across page headings in-stream

```
(TEDIT.HARDCOPY-COLUMN-END LINES ORPHAN FORCENEXTPAGE CHNO PAGEFOOTNOTES REGION TEXTOBJ
  FORMATTINGSTATE FINAL-CHNO)))
```

(TEDIT.FORMATFOLIO

```
[LAMBDA (TEXTOBJ PRSTREAM FORMATTINGSTATE PAGEREGION)
```

; Edited 15-Mar-2024 19:24 by rmk
 ; Edited 13-Mar-2024 09:00 by rmk
 ; Edited 19-Jan-2024 23:28 by rmk
 ; Edited 18-Jan-2024 17:04 by rmk
 ; Edited 13-Nov-2023 00:24 by rmk
 ; Edited 1-Jun-2023 00:12 by rmk
 ; Edited 9-May-2023 21:39 by rmk
 ; Edited 30-May-91 12:51 by jds

:: Print a page number (called a "folio" in the biz) at the location and with the alignment specified in the REGIONSPEC.

```
(LET ((REGION (SCALEREGION (DSPSCALE NIL PRSTREAM)
                          (fetch (PAGEREGION REGIONSPEC) of PAGEREGION)))
      (FOLIOINFO (fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION))
      FOLIOSTREAM FOLIOTEXTOBJ PAGE# FOLIOFORMAT PRETEXT POSTTEXT INFOLIST)
      (CL:UNLESS (AND (GETPFS FORMATTINGSTATE FIRSTPAGE)
                    (LISTGET FOLIOINFO 'NOFIRSTPAGE))
                ; If this isn't the first page, OR we want a page # on the first
                ; page, go ahead and format it.
                ; A LIST OF (FORMAT PRETEXT POSTTEXT)
                (SETQ INFOLIST (LISTGET FOLIOINFO 'FORMATINFO))
                (SETQ FOLIOFORMAT (CAR INFOLIST))
                (SETQ PRETEXT (OR (CADR INFOLIST)
                                  ""))
                (SETQ POSTTEXT (OR (CADDR INFOLIST)
                                   ""))
                [SETQ PAGE# (COND
                          ((GETPFS FORMATTINGSTATE PAGE#TEXT)
                           (MKSTRING (GETPFS FORMATTINGSTATE PAGE#TEXT)))
                          (T (SELECTQ FOLIOFORMAT
                                     (LOWERROMAN (ROMANNUMERALS (GETPFS FORMATTINGSTATE PAGE#)))
                                     (UPPERROMAN (ROMANNUMERALS (GETPFS FORMATTINGSTATE PAGE#)
                                                                T))
                                     (MKSTRING (GETPFS FORMATTINGSTATE PAGE#))
```

:: Bind the stream to make sure it isn't collected.

```
[SETQ FOLIOSTREAM (OPENTEXTSTREAM NIL NIL NIL NIL '(PARALOOKS , (LISTGET FOLIOINFO 'PARALOOKS)
                                                    LOOKS
                                                    , (LISTGET FOLIOINFO 'CHARLOOKS]
(SETQ FOLIOTEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of FOLIOSTREAM))
(TEDIT.INSERT FOLIOTEXTOBJ (CONCAT PRETEXT PAGE# POSTTEXT)
  1 NIL T)
```

```
(bind LINE YBOT FORCENEXTPAGE (TEXTLEN _ (TEXTLEN FOLIOTEXTOBJ))
(BOTTOM _ (fetch (REGION BOTTOM) of REGION))
(CHNO _ 1) while (ILEQ CHNO TEXTLEN) until FORCENEXTPAGE
collect (SETQ LINE (\TEDIT.FORMATLINE FOLIOTEXTOBJ CHNO NIL REGION PRSTREAM FORMATTINGSTATE))
(SETQ FORCENEXTPAGE (EQ (CHARCODE FORM)
(GETLD LINE FORCED-END)))
; Format the next possible line
(SETQ CHNO (ADD1 (FGETLD LINE LCHARLIM))) ; Keep track of the next character...
[SETQ YBOT (COND
(YBOT ; We're into it; take account of this line's height
(IDIFFERENCE YBOT (FGETLD LINE LHEIGHT)))
(T ; Just starting out; find the line's position with respect to the top
; of the region to be filled.
(IDIFFERENCE BOTTOM (FGETLD LINE DESCENT))
(CL:WHEN (ILESSP YBOT (IDIFFERENCE BOTTOM (FGETLD LINE DESCENT)))
(GO $$ITERATE))
(SETYPOS LINE YBOT) ; This line is still good
LINE))] ]]
```

(\TEDIT.FORMAT.FOUNDBOX?

```
[LAMBDA (PAGEREGION FORMATTINGSTATE) ; Edited 19-Jan-2024 23:34 by rmk
; Edited 2-Jul-2023 19:07 by rmk
; Edited 19-Apr-88 17:35 by jds
```

;;; Return T if we're either not looking to begin in a new box, or we are and we've found it.

;;; This is part of generalizing the 'go to a new page' code to allow going to an arbitrary new formatting box.

```
(SELECTQ (GETPFS FORMATTINGSTATE STATE) ; we're just munching along formatting. Keep going.
(FORMATTING T)
(SEARCHING ; We're searching for a page box of the right type. Decide if this
; is it or not.
(CL:WHEN (EQ (GETPFS FORMATTINGSTATE REQUIREDREGIONTYPE)
(fetch (PAGEREGION REGIONTYPE) of PAGEREGION))
; What we're looking for matches what we've got. Turn off the
; search and return T
(SETPFS FORMATTINGSTATE STATE 'FORMATTING)
T))
(:SEARCHING-FOR-EQUIVALENT-PAGE ; We've switched document formats in mid-document, and need to find the corresponding page
; frame to continue properly.
(CL:WHEN (IEQP (GETPFS FORMATTINGSTATE REQUIREDREGIONTYPE)
(GETPFS FORMATTINGSTATE PAGECOUNT)) ; We've formatted enough pages up to now.
(SETPFS FORMATTINGSTATE STATE 'FORMATTING)))
T))
```

;; Aux function to capture page headings during line formatting:

```
(DEFINEQ
```

(\TEDIT.HARDCOPY.PAGEHEADINGS

```
[LAMBDA (TEXTOBJ CHNO FORMATTINGSTATE) ; Edited 17-Mar-2024 00:27 by rmk
; Edited 15-Mar-2024 13:54 by rmk
; Edited 9-May-2023 17:46 by rmk
; Edited 7-May-2023 23:45 by rmk
; Edited 9-Oct-2022 17:12 by rmk
```

;; This runs thru all the headings starting at CHNO, copying the pieces of the different heading types into FORMATTINGSTATE, and returning the
;; starting CHNO of the first non-heading piece.

```
(CL:UNLESS FORMATTINGSTATE ; If it isn't there, we would loose the headings
(SHOULDNT "NIL FORMATTINGSTATE"))
(bind HEADINGSUBTYPE (PC _ (\TEDIT.CHTOPC CHNO TEXTOBJ)) while [AND PC (EQ 'PAGEHEADING (fetch FMTPARATYPE
of (PPARALOOKS PC]
do (SETQ HEADINGSUBTYPE (fetch FMTPARASUBTYPE of (PPARALOOKS PC)))
(for P (START _ CHNO)
inpieces PC while [AND (EQ 'PAGEHEADING (fetch FMTPARATYPE of (PPARALOOKS P)))
(EQ HEADINGSUBTYPE (fetch FMTPARASUBTYPE of (PPARALOOKS P]
do ; We loop at least once, because P=PC satisfies the while. We need the CHNO, not the piece for the piecerange
(add CHNO (PLEN P))
finally (LISTPUT (fetch PAGEHEADINGS of FORMATTINGSTATE)
HEADINGSUBTYPE
(\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES START CHNO TEXTOBJ)))
; Set PC to continue looking for the next headingtype.
(SETQ PC P))
CHNO))
```

;; Aux function to handle end-of-column processing (paragraph keep, widow elimination, etc):

(DEFINEQ

(TEDIT.HARDCOPY-COLUMN-END

[LAMBDA (ORIGINAL-LINES ORPHAN FORCENEXTPAGE CHNO FOOTNOTELINES REGION TEXTOBJ FORMATTINGSTATE FINAL-CHNO
DONT-KEEP-SINGLE-LINE)

; Edited 19-Jan-2024 23:30 by rmk
; Edited 29-Oct-2023 18:40 by rmk
; Edited 4-Jul-2023 21:00 by rmk
; Edited 15-Feb-2023 23:45 by rmk
; Edited 3-Oct-2022 18:08 by rmk
; Edited 11-May-93 01:21 by jds

:: Do column-end processing for TEdit hardcopy -- widow elimination, respect keep-together specifications, etc.

:: RETURNS:

:: -- List of line descriptors in the column
:: -- List of line descriptors removed from the end of the column.
:: -- Flag to say that it kept one line

(SETQ ORIGINAL-LINES (DREMOVE NIL ORIGINAL-LINES))

; Remove any NILs from the list of lines; they're artifacts of
; running into page headings in mid-page.

(LET ((LINES (COPY ORIGINAL-LINES))
LASTLINE
(REMOVED-LINES (LIST ORPHAN)))
(CL:WHEN LINES

; Only worry about widows and orphans if there are really lines to
; worry about

[for LINE in LINES when (GETLD LINE LMARK LINE) do

(DREMOVE LINE LINES)
(SETQ FINAL-CHNO
(AND FINAL-CHNO (IMAX FINAL-CHNO
(ADD1 (GETLD LINE LCHARLIM

(SETQ LASTLINE (CAR (FLAST LINES)))
(CL:WHEN (AND ORPHAN (GETLD ORPHAN LSTLN)
(NOT (GETLD ORPHAN 1STLN)))

; Find the last line in this box (column or page)

:: There was an overhanging line, and it was the last line of the paragraph. Remove the penultimate line.

(SETQ LINES (DREMOVE LASTLINE LINES))
(PUSH REMOVED-LINES LASTLINE)
(SETQ FINAL-CHNO (GETLD LASTLINE LCHAR1))
(SETQ LASTLINE (CAR (FLAST LINES)))
(CL:WHEN (AND LASTLINE (GETLD LASTLINE 1STLN)
(NOT (GETLD LASTLINE LSTLN))
(ILESSP (GETLD LASTLINE LCHARLIM)
(TEXTLEN TEXTOBJ)))

; The last line on the page is a widow. Remove it, too.

(SETQ LINES (DREMOVE LASTLINE LINES))
(PUSH REMOVED-LINES LASTLINE)
(SETQ FINAL-CHNO (GETLD LASTLINE LCHAR1))
(SETQ LASTLINE (CAR (FLAST LINES)))

[COND
[(NOT LINES)

:: This is a 2- or 3-line paragraph, with only the first 1 or 2 lines fitting on ANY page. Just return the first 1 or two lines, and we'll
:: have to eat the widow.

(SETQ LINES ORIGINAL-LINES)
(SETQ FINAL-CHNO (COND
(ORPHAN (GETLD ORPHAN LCHAR1))
(T (ADD1 (GETLD (CAR (FLAST ORIGINAL-LINES))
LCHARLIM)

([AND (NEQ FORCENEXTPAGE 'USERBREAK)
(ILEQ CHNO (TEXTLEN TEXTOBJ))
(OR (fetch (FMTSPEC FMTHEADINGKEEP) of (GETLD LASTLINE LFMTSPEC))
(AND (fetch (FMTSPEC FMTKEEP) of (GETLD LASTLINE LFMTSPEC))
(NOT (GETLD LASTLINE LSTLN]

:: Only do widow/orphan detection if this is NOT a page break the user asked for. And this isn't the end of the document.

(for LASTLINE in (REVERSE LINES) while [OR (fetch (FMTSPEC FMTHEADINGKEEP) of (GETLD LASTLINE
LFMTSPEC))
(AND (fetch (FMTSPEC FMTKEEP) of (GETLD LASTLINE
LFMTSPEC))
(NOT (GETLD LASTLINE LSTLN]

do ; Run thru, removing any trailing headings. However, assure that there's at least one line on a page.

finally (COND
(AND LASTLINE (AND (NOT (fetch (FMTSPEC FMTHEADINGKEEP) of (GETLD LASTLINE LFMTSPEC))
(GETLD LASTLINE LSTLN)))

:: OK we found a line that DOESN'T need to be kept with the other paragraphs. Chop off the list starting AFTER
:: it.

(SETQ LASTLINE (CDR (MEMB LASTLINE LINES)))
(SETQ LINES (LDIFF LINES LASTLINE))
(SETQ REMOVED-LINES (APPEND LASTLINE REMOVED-LINES))
(SETQ FINAL-CHNO (GETLD (CAR LASTLINE)
LCHAR1))

(T (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "WARNING: Page " (GETPFS FORMATTINGSTATE
PAGE#)
" is completely full of headings ")


```

    T])
  (CL:WHEN FOOTNOTELINES
    ;; There are footnotes--fix up their vertical locations, so they're aligned on the botton of the column.
    (bind (YBOT _ (IDIFFERENCE (fetch (REGION BOTTOM) of REGION)
      (GETLD (CAR (FLAST FOOTNOTELINES))
        DESCENT)))
      for LINE in (REVERSE FOOTNOTELINES) do (SETYPOS LINE YBOT)
        (add YBOT (FGETLD LINE LHEIGHT))))
  (COND
    ((OR LINES FOOTNOTELINES) ; There really ARE lines in this column; take care of them.
      (TEDIT.VALUES (APPEND LINES FOOTNOTELINES)
        REMOVED-LINES NIL))
    [(AND ORPHAN (NOT ORIGINAL-LINES)
      (NOT DONT-KEEP-SINGLE-LINE)) ; If there's only one line left for this box, return it anyhow.
      (TEDIT.VALUES (CONS ORPHAN FOOTNOTELINES)
        NIL
        (ADD1 (GETLD ORPHAN LCHARLIM)
          (AND (NOT DONT-KEEP-SINGLE-LINE)
            REMOVED-LINES)
          (SETQ LASTLINE (CAR REMOVED-LINES))
          (TEDIT.VALUES (LIST LASTLINE)
            (CDR REMOVED-LINES)
            (AND LASTLINE (ADD1 (GETLD LASTLINE LCHARLIM)
              (ORPHAN
                ;; There's only the one line, so let's go back and try again.
                (TEDIT.VALUES NIL (LIST ORPHAN)
                  FINAL-CHNO]))
              )
            )
          )
        )
    )
  )

```

;; Handle varying paper sizes

(DEFINEQ

(SCALEPAGEUNITS

[LAMBDA (VALUE FACTOR PAPERSIZE) (* jds "14-Jun-85 15:34")

(* Scale a page-relative value into points%: Scale VALUE by FACTOR, then allow for negative values to mean "come in from the other side by that much")

```

  (AND VALUE (PROG [(TVAL (FIXR (FTIMES VALUE FACTOR)))
    (OTHEREDGE (SELECTQ PAPERSIZE
      (NIL LETTER)
      612)
      (LEGAL 612)
      (fetch (TEDITPAPERSIZE TPSWIDTH) of (ASSOC PAPERSIZE TEDIT.PAPER.SIZES]
    [COND
      ((ILESSP TVAL 0) (* He specified this value as an offset from the opposite edge.
        Convert it.)
        (SETQ TVAL (IPLUS OTHEREDGE TVAL])
      (RETURN TVAL])
    )
  )

```

(SCALEPAGEXUNITS

[LAMBDA (VALUE FACTOR PAPERSIZE LANDSCAPE?) ; Edited 21-Apr-88 10:46 by jds

;; Scale a page-relative value into points: Scale VALUE by FACTOR, then allow for negative values to mean 'come in from the other side by that
;; much'

```

  (AND VALUE (PROG [(TVAL (FIXR (FTIMES VALUE FACTOR)))
    OTHEREDGE)
    [COND
      ((ILESSP TVAL 0) ; He specified this value as an offset from the opposite edge.
        ; Convert it.
        (SETQ OTHEREDGE (\TEDIT.PAPERWIDTH PAPERSIZE LANDSCAPE?))
        (SETQ TVAL (IPLUS OTHEREDGE TVAL])
      (RETURN TVAL])
    )
  )

```

(SCALEPAGEYUNITS

[LAMBDA (VALUE FACTOR PAPERSIZE LANDSCAPE?) ; Edited 17-Dec-87 14:52 by jds

;; Scale a page-relative value into points: Scale VALUE by FACTOR, then allow for negative values to mean 'come in from the other side by that
;; much'

```

  (AND VALUE (PROG [(TVAL (FIXR (FTIMES VALUE FACTOR)))
    OTHEREDGE)
    [COND
      ((ILESSP TVAL 0) ; He specified this value as an offset from the opposite edge.
        ; Convert it.
        (SETQ OTHEREDGE (\TEDIT.PAPERHEIGHT PAPERSIZE LANDSCAPE?))
        (SETQ TVAL (IPLUS OTHEREDGE TVAL])
      (RETURN TVAL])
    )
  )

```

(\TEDIT.PAPERHEIGHT

[LAMBDA (PAPERSIZE LANDSCAPE?) ; Edited 29-Dec-86 15:06 by jds

::: Compute the HEIGHT of a sheet of paper, according to PAPERSIZE, in points.

```
(COND
  (LANDSCAPE? ; The paper is landscape, so its height is the WIDTH of the same
                ; paper size, normal.
    (\TEDIT.PAPERWIDTH PAPERSIZE NIL))
  (T ; Not landscape, so look up the size spec:
    (SELECTQ PAPERSIZE
      ((NIL LETTER Letter) 792)
      ((Legal 8.5x14 LEGAL) 1008)
      ((A4 a4) 842)
      (fetch (TEDITPAPERSIZE TPSHEIGHT) of (ASSOC PAPERSIZE TEDIT.PAPER.SIZES]))
```

(\TEDIT.PAPERWIDTH

```
[LAMBDA (PAPERSIZE LANDSCAPE?) ; Edited 9-Dec-87 20:10 by jds
```

::: Compute the WIDTH of a sheet of paper, according to PAPERSIZE and LANDSCAPE?

```
(LET (CANONICAL-PAPERSIZE)
  (COND
    (LANDSCAPE? ; It's landscape paper, so look at the HEIGHT of the
                 ; corresponding normal paper.
      (\TEDIT.PAPERHEIGHT PAPERSIZE NIL))
    (T ; Not landscape, so look up the size spec:
      (SELECTQ PAPERSIZE
        ((NIL Letter LETTER 8.5x11) 612) ; letter size paper, 8.5inx11in
        ((Legal LEGAL 8.5x14) 612)
        ((A4 a4) 595) ; A4 ISO-size paper, 210mmx297mm
        (COND
          ((SETQ CANONICAL-PAPERSIZE (ASSOC PAPERSIZE TEDIT.PAPER.SIZES))
            (fetch (TEDITPAPERSIZE TPSWIDTH) of CANONICAL-PAPERSIZE))
          (T (\ILLEGAL.ARG PAPERSIZE)))
```

)

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS TEDIT.PAPER.SIZES)
)
```

(RPAQQ TEDIT.PAPER.SIZES

```
((A0 2384 3370)
 (A1 1684 2384)
 (A2 1191 1684)
 (A3 842 1191)
 (A4 595 842)
 (A5 420 595)
 (B0 2835 4008)
 (B1 2004 2835)
 (B2 1417 2004)
 (B3 1001 1417)
 (B4 709 1001)
 (B5 499 709))
```

::: Page numbering option support

```
(DEFINEQ
```

(ROMANNUMERALS

```
[LAMBDA (NUMBER UCFLG) (* jds "12-Jul-85 13:19")
```

(* * Take a NUMBER, and render it as a string of roman numerals.
 If UCFLG, then the numerals will be upper-case; otherwise, they are lower-case.)

```
(PROG ((CHARS NIL))
  [while (NOT (ZEROP NUMBER)) do (COND
    ((IGEQ NUMBER 1000)
      (push CHARS 'm)
      (add NUMBER -1000))
    ((IGEQ NUMBER 900)
      (push CHARS 'c)
      (push CHARS 'm)
      (add NUMBER -900))
    ((IGEQ NUMBER 500)
      (push CHARS 'd)
      (add NUMBER -500))
    ((IGEQ NUMBER 400)
      (push CHARS 'c)
      (push CHARS 'd)
```

```

(add NUMBER -400)
((IGE Q NUMBER 100)
(push CHARS 'c)
(add NUMBER -100)
((IGE Q NUMBER 90)
(push CHARS 'x)
(push CHARS 'c)
(add NUMBER -90)
((IGE Q NUMBER 50)
(push CHARS 'l)
(add NUMBER -50)
((IGE Q NUMBER 40)
(push CHARS 'x)
(push CHARS 'l)
(add NUMBER -40)
((IGE Q NUMBER 10)
(push CHARS 'x)
(add NUMBER -10)
((IGE Q NUMBER 9)
(push CHARS 'i)
(push CHARS 'x)
(add NUMBER -9)
((IGE Q NUMBER 5)
(push CHARS 'v)
(add NUMBER -5)
((IGE Q NUMBER 4)
(push CHARS 'i)
(push CHARS 'v)
(add NUMBER -4)
(T (push CHARS 'i)
(add NUMBER -1]

```

```

(RETURN (COND
  [UCFLG
    (U-CASE (CONCATLIST (REVERSE CHARS]
    (T (CONCATLIST (REVERSE CHARS])

```

(* The caller wants his roman numerals upper case)

)

:: Foot note support

(DEFINEQ

(\TEDIT.FORMAT.FOOTNOTE

```

[LAMBDA (TEXTOBJ PRSTREAM LINE REGION FORMATTINGSTATE)

```

```

; Edited 15-Mar-2024 19:24 by rmk
; Edited 13-Mar-2024 17:00 by rmk
; Edited 19-Jan-2024 23:30 by rmk
; Edited 6-May-2023 20:38 by rmk
; Edited 7-Mar-2023 13:11 by rmk
; Edited 30-May-91 12:52 by jds

```

```

;; Grab text from the TEXTOBJ, starting with CHNO, and use it to fill REGION on a page. Return a list of line descriptors which, taken together, fill
;; the region.

```

(BLOCK)

; Footnotes aren't so long, but why not?

```

(bind PREVLINE (LEFT _ (fetch (REGION LEFT) of REGION))

```

```

(TEXTLEN _ (TEXTLEN TEXTOBJ))

```

```

(CHNO _ (GETLD LINE LCHAR1)) while (ILEQ CHNO TEXTLEN) until (AND PREVLINE (GETLD PREVLINE LSTLN))

```

collect

:: Grab a line descriptor from the formatting list, or create a new one.

```

(SETQ LINE (\TEDIT.FORMATLINE TEXTOBJ CHNO (GETPFS FORMATTINGSTATE PAGELINECACHE)
  REGION PRSTREAM FORMATTINGSTATE)) ; Format the line, noting any form-feeds

```

```

(add (FGETLD LINE LEFTMARGIN)

```

```

LEFT)

```

```

(add (FGETLD LINE RIGHTMARGIN)

```

```

LEFT)

```

```

; Format the next possible line
; Keep track of the next character...

```

```

(SETQ CHNO (ADD1 (FGETLD LINE LCHARLIM)))

```

```

(SETQ PREVLINE LINE)

```

```

LINE

```

finally

```

; Remove any NILs from the line list; they're artifacts of running
; across page headings in-stream

```

```

(RETURN (DREMOVE NIL $$VAL])

```

)

FUNCTION INDEX

ROMANNUMERALS	18	TEDIT.PAGEFORMAT	5	\TEDIT.FORMATPAGE	10
SCALEPAGEUNITS	17	TEDIT.SINGLE.PAGEFORMAT	4	\TEDIT.FORMATTTEXTBOX	12
SCALEPAGEXUNITS	17	TEDIT.SKIP.SPECIALCOND	8	\TEDIT.PAPERHEIGHT	17
SCALEPAGEYUNITS	17	\TEDIT.FORMAT.FOOTNOTE	19	\TEDIT.PAPERWIDTH	18
TEDIT.COMPOUND.PAGEFORMAT	5	\TEDIT.FORMAT.FOUNDBOX?	15	\TEDIT.PARSE.PAGEFRAMES	3
TEDIT.FORMAT.HARDCOPY	6	\TEDIT.FORMATBOX	8	\TEDIT.PUT.PAGEFRAMES	3
TEDIT.HARDCOPY-COLUMN-END	16	\TEDIT.FORMATFOLIO	14	\TEDIT.UNPARSE.PAGEFRAMES	3
TEDIT.HARDCOPY.PAGEHEADINGS	15	\TEDIT.FORMATHEADING	10		

VARIABLE INDEX

MAXPAGE#	3	TEDIT.DEFAULT.FOLIO.LOOKS	3	TEDIT.PAPER.SIZES	18
MINPAGE#	3	TEDIT.PAGE.FRAMES	3		

MACRO INDEX

GETPFS	2	TEDIT.SETQS	2	\FIRST-COLUMN-START	2
SETPFS	2	TEDIT.VALUES	3		

RECORD INDEX

PAGEFORMATTINGSTATE	2	PAGEREGION	2	TEDITPAPERSIZE	1
---------------------------	---	------------------	---	----------------------	---
