

File created: 23-Feb-2025 10:06:16 {WMEDLEY}<library>TEDIT>TEDIT-PAGE.;208

edit by: rmk

changes to: (FNS TEDIT.FORMAT.HARDCOPY)

previous date: 19-Feb-2025 13:33:12 {WMEDLEY}<library>TEDIT>TEDIT-PAGE.;207

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 (\TEDIT.CHARLOOKS.FROM.FONT (FONTCOPY NIL
    ' (FAMILY MODERN SIZE 10
      WEIGHT MEDIUM
      SLOPE REGULAR]

(VARS (MAXPAGE# 65535)
  (MINPAGE# 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 TEDIT.GET.PAGEFORMAT)
  (INITVARS (TEDIT.PAGE.FRAMES (TEDIT.COMPOUND.PAGEFORMAT (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]

(FNS TEDIT.FORMAT.HARDCOPY)
[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? \TEDIT.SKIP.SPECIALCOND)
  ;; 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
  (FNS ROMANNUMERALS)) ; Page numbering option support
[COMS
  (FNS TEDIT.PAGENO.CREATE \TEDIT.PAGENO.OBJINIT \TEDIT.PAGENO.BUTTONEVENTINFN
    \TEDIT.PAGENO.IMAGEBOXFN \TEDIT.PAGENO.DISPLAYFN \TEDIT.PAGENO.GETFN \TEDIT.PAGENO.PUTFN)
  (P (\TEDIT.PAGENO.OBJINIT))) ; Page number image obj
[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.

```

))

)

:: FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(RECORD PAGEFORMATTINGSTATE (;; Contains the state for a TEdit page-formatting job.

```

PAGE# ; The current page number. Counted from 1
FIRSTPAGE
;; T if the current page is the 'first page'. Is set initially, and can be set again by the user at will. Gets reset after
;; each page image is printed.
MINPAGE# ; The page # of the first page to be printed, or NIL
MAXPAGE# ; The page # of the last page to be printed, or NIL
STATE ; One of FORMATTING or SEARCHING.
REQUIREDREGIONTYPE ; If STATE is SEARCHING, the kind of box we're looking for. If
STATE is :SEARCHING-FOR-EQUIVALENT-PAGE, this is the
page count for the matching page.
MAINSTREAM ; The principal textobj/stream source
CHNO ; Our position in that stream
PRESSREGION ; The press code's REGION info.
PAGEHEADINGS ; The list of current values to be printed, indexed by heading type
PAGE#GENERATOR ; List of page numbers; later, maybe, a function to generate
page numbers. Used to fill in PAGE#TEXT, below
PAGE#TEXT ; If special page numbers are in use, this is the place to take
them from. PAGE# is still used for recto/verso decisions &
PAGEISRECTO ; T if this is a recto page, NIL if it's a VERSO page.
PAGEFOOTNOTELINES ; A list of extant footnote lines that should appear at the next
opportunity
PAGEFLOATINGTOPLINES ; A list of lines that should float to the top of the next available
place
PAGECOUNT ; The number of pages we've formatted so far.
PAGELINECACHE ; A cache for pre-created LINEDESCRIPTOR/THISLINE sets, to
avoid the overhead of re-allocating them all the time
NEWPAGELAYOUT ; If we switch page layouts in mid-document, this is where the
new layout gets cached until we get started again.

```

)

PAGECOUNT _ 0)

(DATATYPE PAGEREGION (;; Describe a part of a page for page formatting. Can be made into compound descriptions.

```

REGIONFILLMETHOD ; What kind of a region this is -- TEXT, FOLIO, PAGEHEADING,
; etc.
REGIONSPEC ; The page-relative region this occupies
REGIONLOCALINFO ; A PLIST for local information
(REGIONPARENT FULLXPOINTER) ; The parent node for this box, for sub-boxes
REGIONSUBBOXES ; The sub-regions of this region
REGIONTYPE ; A user-settable region type

```

))

)

(/DECLAREDATATYPE 'PAGEREGION ' (POINTER POINTER POINTER FULLXPOINTER POINTER POINTER)

;; ---field descriptor list elided by lister---
' 12)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETPFS MACRO ((FS FIELD)
(fetch (PAGEFORMATTINGSTATE FIELD) of FS)))

(PUTPROPS SETPFS MACRO ((FS FIELD NEWVALUE)
(replace (PAGEFORMATTINGSTATE FIELD) of FS with NEWVALUE)))

)

:: END EXPORTED DEFINITIONS

(DECLARE%: EVAL@COMPILE

(PUTPROPS \FIRST-COLUMN-START MACRO [(LINE PARALOOKS)
(AND (FGETLD LINE 1STLN)
(EQ 'FIRST (FGETPLOOKS PARALOOKS FMTCOLUMN])

)

(DECLARE%: EVAL@COMPILE

(PUTPROPS TEDIT.SETQS MACRO [ARGS `(LET ((\$VALUES , (CADR ARGS))

```

($$PRIMARY))
(DECLARE (LOCALVARS $$VALUES))
(PROG1 (CAR $$VALUES)
,@[FOR V IN (CAR ARGS) collect (COND
[V `(SETQ ,V (POP $$VALUES)
(T `(SETQ $$VALUES (CDR $$VALUES)
])

```

```

(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 [\TEDIT.CHARLOOKS.FROM.FONT (FONTCOPY NIL
' (FAMILY MODERN SIZE 10 WEIGHT MEDIUM SLOPE
REGULAR])

```

```

(RPAQQ MAXPAGE# 65535)
(RPAQQ MINPAGE# 1)

```

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

```

(DEFINEQ

```

(\TEDIT.PARSE.PAGEFRAMES

```

[LAMBDA (PAGELIST PARENT)

```

```

; Edited 30-Aug-2024 15:40 by rmk
; 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 (SETQ PAGELIST (CAR PAGELIST))
(TEDIT.COMPOUND.PAGEFORMAT (\TEDIT.PARSE.PAGEFRAMES (pop PAGELIST))
(\TEDIT.PARSE.PAGEFRAMES (pop PAGELIST))
(\TEDIT.PARSE.PAGEFRAMES (pop PAGELIST))

```

(\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 10-Jan-2025 11:41 by rmk
; Edited 24-Dec-2024 21:20 by rmk
; Edited 15-Aug-2024 23:01 by rmk
; Edited 6-Aug-2024 12:06 by rmk
; 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. This is a documented user entry, and the user gets to
 :: decide the units of the various specified parameters. But internally everything is stored as points, and oddly, the page layout menu doesn't reflect
 :: the original units, only picas.

```
(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)
(SETQ SCALEFACTOR (SELECTQ UNITS
(PPOINTS NIL) ; If units are in printers points, the default, do no scaling
1)
(PICAS PTSPERPICA)
(INCHES PTSPERINCH)
(MICAS PTSPERMICA)
(CM PTSPERCM)
(\ILLEGAL.ARG UNITS)))
(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 PAGEREGION (IDIFFERENCE (IDIFFERENCE PAPERWIDTH RIGHT)
LEFT))
(CL:WHEN PAGE#S?
```

:: This asserts that the page number's region is 4 inches wide. Why? What if the pretext/posttext is longer?

```
(SELECTQ (U-CASE 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 (ITIMES 4 PTSPERINCH))))
((CENTERED CENTER NIL) ; Otherwise, center the page number around the point he
; specifies
(SETQ FOLIOLEFT (IDIFFERENCE PX (ITIMES 2 PTSPERINCH))))
(ERROR "Invalid page number alignment" PQUAD))
```

:: 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.

:: RMK: Very odd to default here 4 inches and 1/2 for the folio region.

:: PY is described as the baseline of the page numbers, measured from the bottom of the page. So the page numbers and
 :: pre/posttext sit above.

```
[SETQ SUBREGIONS (LIST (create PAGEREGION
REGIONFILLMETHOD _ 'FOLIO
REGIONSPEC _ (create REGION
LEFT _ FOLIOLEFT
BOTTOM _ PY
WIDTH _ (ITIMES 4 PTSPERINCH)
HEIGHT _ (IQUOTIENT PTSPERINCH 2))
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 when (CAR HDG)
  collect
  ;; Run thru the list of headings, building a box for each. By default the heading's width runs up to the
  ;; right margin on the page. X/LEFT is the left end of the top line, Y is the "position of the top
  ;; line"--it's YTOP, baseline, or YBOT? But SPECIALX and SPECIALY are described as "the
  ;; distances from the lower-left corner of the paper: the lower-left corner of the paragraph's top line is
  ;; placed at the specified position, so this suggests YBOT.
  (if (AND (NUMBERP (CADR HDG))
           (NUMBERP (CADDR HDG)))
      then (SETQ LEFT (SCALEPAGEXUNITS (CADR HDG)
                                       SCALEFACTOR PAPERWIDTH LANDSCAPE?))
          (create PAGEREGION
                  REGIONFILLMETHOD _ 'HEADING
                  REGIONSPEC _ (create REGION
                                       LEFT _ LEFT
                                       BOTTOM _ (SCALEPAGEYUNITS
                                               (CADR HDG)
                                               SCALEFACTOR PAPERWIDTH
                                               LANDSCAPE?)
                                       WIDTH _ (IMAX (IDIFFERENCE
                                                    PAPERWIDTH
                                                    LEFT)
                                               PTSPERINCH)
                                       HEIGHT _ (QUOTIENT PTSPERINCH 2)
                  )
          REGIONLOCALINFO _ (LIST 'HEADINGTYPE (CAR HDG)))
      else (ERROR (CONCAT "Invalid X/Y position for heading-type "
                          (CAR HDG]
              (SETQ SUBREGIONS (APPEND SUBREGIONS HEADINGREGIONS)))
[COND
 [(OR (NULL COLS)
      (EQ COLS 1))
      ; There is a single column, so treat it as just one text region
      ; bounded by the page margins.
      (SETQ SUBREGIONS (NCONC1 SUBREGIONS
                            (create PAGEREGION
                                    REGIONFILLMETHOD _ 'TEXT
                                    REGIONSPEC _
                                    (create REGION
                                            LEFT _ LEFT
                                            BOTTOM _ BOTTOM
                                            WIDTH _ PAPERWIDTH
                                            HEIGHT _ (IDIFFERENCE (IDIFFERENCE PAPERHEIGHT TOP)
                                                                    BOTTOM]
                            )
      ; There are several columns. We need to create a text box for
      ; each col.
      (CL:UNLESS (OR COLWIDTH INTERCOL)
                 (ERROR "Can't default both Col width and spacing"))
[COND
 [(NULL COLWIDTH)
      ; He wants us to fill in the column width, given margins and
      ; intercolumn spacing.
      (SETQ COLWIDTH (FIXR (FQUOTIENT (IDIFFERENCE PAPERWIDTH (ITIMES INTERCOL (SUB1 COLS)))
                                     COLS]
      (NULL INTERCOL)
      ; Or else he wants to give us just the col width and have us calc
      ; the spacing.
      (SETQ INTERCOL (FIXR (FQUOTIENT (IDIFFERENCE PAPERWIDTH (ITIMES COLWIDTH COLS))
                                     (SUB1 COLS]
      (SETQ SUBREGIONS
            (NCONC SUBREGIONS
                  (for COL from 1 to COLS as CLEFT from LEFT by (IPLUS COLWIDTH INTERCOL)
                    collect (create PAGEREGION
                                    REGIONFILLMETHOD _ 'TEXT
                                    REGIONSPEC _
                                    (create REGION
                                            LEFT _ CLEFT
                                            BOTTOM _ BOTTOM
                                            WIDTH _ COLWIDTH
                                            HEIGHT _ (IDIFFERENCE (IDIFFERENCE PAPERHEIGHT TOP)
                                                                    BOTTOM]
                    )
      (replace REGIONSUBBOXES of PAGEREGION with SUBREGIONS)
      PAGEREGION])

```

(TEDIT.COMPOUND.PAGEFORMAT

[LAMBDA (FIRST VERSO RECTO)

; Edited 20-Jan-2024 12:07 by rmk
; Edited 16-Jan-2024 14:24 by rmk
(* jds "27-Jul-84 10:15")

:: This creates a 2-element SEQUENCE pageformat consisting of FIRST followed by an ALTERNATE pageformat containing VERSO and RECTO

(CL:UNLESS VERSO (SETQ VERSO FIRST))
(CL:UNLESS RECTO (SETQ RECTO VERSO))

```
(create PAGEREGION
  REGIONFILLMETHOD _ 'SEQUENCE
  REGIONSUBBOXES _ (LIST FIRST (create PAGEREGION
    REGIONFILLMETHOD _ 'ALTERNATE
    REGIONSUBBOXES _ (LIST VERSO RECTO)
    REGIONSPEC _ (LIST 0 0 0 0)))

  REGIONSPEC _ (LIST 0 0 0 0])
```

(TEDIT.PAGEFORMAT

```
[LAMBDA (TSTREAM FORMAT PAGEID)
```

```
; Edited 22-Sep-2024 18:45 by rmk
; Edited 3-Sep-2024 22:35 by rmk
; Edited 30-Aug-2024 23:33 by rmk
; Edited 12-Aug-2024 01:10 by rmk
; Edited 4-Feb-2024 22:10 by rmk
; Edited 16-Jan-2024 14:25 by rmk
; Edited 21-Dec-2023 12:32 by rmk
; Edited 12-Jun-90 19:13 by mitani
```

```
:: Programmatic interface for page formatting. If FORMAT is a single page format, it is applied only to PAGETYPE if not NIL. If NIL, it becomes the
:: FIRST/DEFAULT, overriding any LEFT or RIGHT that may already be there. If FORMAT is a composite, then PAGETYPE must be NIL.
:: PAGETYPE argument was not documented. But this preserves the original semantics: a single format with no PAGETYPE is taken as the first
:: and default, wipes out whatever else that might have been there.
:: Original code allowed (undocumented) list of 3 items to be passed around
:: FORMAT can also be another textstream, in which case its format is take as FORMAT--essentially the copy case.
```

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
```

```
:: Note that PAGETYPE and the textstream-format case are extensions, not in the original documentation.
```

```
(LET* ((TEXTOBJ (FGETTSTR TSTREAM TEXTOBJ))
  (OLDPAGEREGION (FGETTOBJ TEXTOBJ TXTPAGEFRAMES))
  (OLDFIRST (TEDIT.GET.PAGEFORMAT OLDPAGEREGION ' |FIRST(&DEFAULT) |))
  OLDLEFT OLDRIGHT NEWPAGEREGION SUBBOXES)
  (CL:WHEN (TEXTSTREAM FORMAT T) ; Get it from somewhere else
    (SETQ FORMAT (GETTOBJ (TEXTOBJ FORMAT)
      TXTPAGEFRAMES)))
  (CL:UNLESS (type? PAGEREGION FORMAT)
    (\ILLEGAL.ARG FORMAT))
  (SETQ NEWPAGEREGION (SELECTQ (fetch (PAGEREGION REGIONFILLMETHOD) of FORMAT)
    (PAGE (SELECTQ PAGEID
      (|FIRST(&DEFAULT) | ; Try to maintain EQ for defaults; use NIL instead?
        (SETQ OLDLEFT (TEDIT.GET.PAGEFORMAT OLDPAGEREGION
          'LEFT))
        (SETQ OLDRIGHT (TEDIT.GET.PAGEFORMAT OLDPAGEREGION
          'RIGHT))
        (TEDIT.COMPOUND.PAGEFORMAT FORMAT (CL:IF (EQ OLDFIRST
          OLDLEFT)
          FORMAT
          OLDLEFT)
          (CL:IF (EQ OLDFIRST OLDRIGHT)
            FORMAT
            OLDRIGHT)))
      (OTHER% LEFT (TEDIT.COMPOUND.PAGEFORMAT OLDFIRST FORMAT
        (TEDIT.GET.PAGEFORMAT OLDPAGEREGION
          'RIGHT)))
      (OTHER% RIGHT (TEDIT.COMPOUND.PAGEFORMAT OLDFIRST
        (TEDIT.GET.PAGEFORMAT OLDPAGEREGION
          'LEFT)
        FORMAT))
      (NIL ; Both LEFT and RIGHT default to FIRST, as indicated by EQ tests
        (TEDIT.COMPOUND.PAGEFORMAT FORMAT FORMAT FORMAT))
      (\ILLEGAL.ARG)))
    (SEQUENCE ; TEDIT.COMPOUND.PAGEFORMAT produces this
      ; complicated arrangement, don't know why
      (SETQ SUBBOXES (fetch (PAGEREGION REGIONSUBBOXES) of FORMAT))
      (CL:UNLESS [AND (EQ 'PAGE (fetch (PAGEREGION REGIONFILLMETHOD)
        of (CAR SUBBOXES)))
        (EQ 'ALTERNATE (fetch (PAGEREGION REGIONFILLMETHOD)
          of (CADR SUBBOXES)))
        (EQ 'PAGE (fetch (PAGEREGION REGIONFILLMETHOD)
          of (CAR (fetch (PAGEREGION
            REGIONSUBBOXES)
            of (CADR SUBBOXES)]
            (EQ 'PAGE (fetch (PAGEREGION REGIONFILLMETHOD)
              of (CADR (fetch (PAGEREGION
                REGIONSUBBOXES)
                of (CADR SUBBOXES)]
                (\ILLEGAL.ARG NEWPAGEREGION))
              FORMAT)
              (\ILLEGAL.ARG FORMAT))
```

```
:: NEWPAGEREGION is now a compound PAGEREGION with default installed, ready to install. .
```

```
(CL:UNLESS (EQUALALL NEWPAGEREGION OLDPAGEREGION) ; This doesn't catch the default relations, which are based on EQ
  ; to first.
  (SETTOBJ TEXTOBJ TXTPAGEFRAMES NEWPAGEREGION) ; If NIL, this must be the call from OPENTEXTSTREAM, no
  ; history or dirt
```

```
(CL:WHEN OLDPAGEREGION
  (\TEDIT.HISTORYADD TEXTOBJ (\TEDIT.HISTORY.EVENT TEXTOBJ :PageFormat NIL NIL NIL NIL
    OLDPAGEREGION))
  (SETTOBJ TEXTOBJ \DIRTY T)))
TSTREAM])
```

(TEDIT.GET.PAGEFORMAT

```
[LAMBDA (TSTREAM PAGETYPE NOERROR) ; Edited 30-Aug-2024 15:19 by rmk
  ;; Returns a single PAGEREGION for PAGETYPE in TSTREAM. Essentially unravels the components that are slapped together by
  ;; TEDIT.COMPOUND.PAGEFORMAT. To get the whole compound format, use GETTEXTPROP.
  ;; This also decodes a list of 3 single pageregions.
```

```
(LET ((PAGEREGION (if (type? PAGEREGION TSTREAM)
  then TSTREAM
  elseif (AND (TEXTOBJ TSTREAM T)
    (FGETTOBJ (TEXTOBJ TSTREAM)
      TXTPAGEFRAMES))
  elseif (LISTP TSTREAM))
  PAGETYPE)
  REST)
  (if (LISTP PAGEREGION)
    then ; Maybe this should be deprecated.
      (SELECTQ PAGETYPE
        ((FIRST DEFAULT |FIRST(&DEFAULT) | FIRST/DEFAULT)
          (CAR PAGEREGION))
        ((OTHER% LEFT LEFT)
          (TEDIT.COMPOUND.PAGEFORMAT NIL PAGEREGION NIL)
          (CADR PAGEREGION))
        ((OTHER% RIGHT RIGHT)
          (CADDR PAGEREGION))
        (CL:UNLESS NOERROR (\ILLEGAL.ARG PAGETYPE)))
      elseif (AND (type? PAGEREGION PAGEREGION)
        (EQ 'SEQUENCE (fetch (PAGEREGION REGIONFILLMETHOD) of PAGEREGION)))
        then (SETQ REST (CADR (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION)))
          (SELECTQ PAGETYPE
            ((FIRST DEFAULT |FIRST(&DEFAULT) | FIRST/DEFAULT)
              (CAR (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION)))
            ((OTHER% LEFT LEFT)
              (CAR (CL:IF (EQ 'ALTERNATE (fetch (PAGEREGION REGIONFILLMETHOD) of REST))
                (fetch (PAGEREGION REGIONSUBBOXES) of REST)
                REST)))
            ((OTHER% RIGHT RIGHT)
              (CADR (CL:IF (EQ 'ALTERNATE (fetch (PAGEREGION REGIONFILLMETHOD) of REST))
                (fetch (PAGEREGION REGIONSUBBOXES) of REST)
                REST)))
            (CL:UNLESS NOERROR (\ILLEGAL.ARG PAGETYPE)))
          elseif PAGEREGION
          then (CL:UNLESS NOERROR (\ILLEGAL.ARG PAGETYPE])
    )
```

(RPAQ? TEDIT.PAGE.FRAMES

```
(TEDIT.COMPOUND.PAGEFORMAT (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)))
```

(DEFINEQ

(TEDIT.FORMAT.HARDCOPY

```
[LAMBDA (TEXTSTREAM FILE DONTSEND BREAKPAGETITLE SERVER PRINTOPTIONS IMAGETYPE FIRSTPG# STARTPG ENDPG QUIET)
  ; Edited 23-Feb-2025 09:59 by rmk
  ; Edited 30-Aug-2024 15:45 by rmk
  ; Edited 10-Jul-2024 23:34 by rmk
  ; Edited 29-Jun-2024 10:32 by rmk
  ; Edited 5-Apr-2024 08:01 by rmk
  ; Edited 16-Mar-2024 09:31 by rmk
  ; Edited 7-Mar-2024 12:34 by rmk
  ; Edited 19-Jan-2024 23:39 by rmk
  ; Edited 24-Dec-2023 14:10 by rmk
  ; Edited 15-Nov-2023 23:56 by rmk
  ; Edited 22-Sep-2023 20:38 by rmk
  ; Edited 4-Jul-2023 11:16 by rmk
  ; Edited 2-Oct-2022 00:00 by rmk
  ; Edited 25-May-93 13:06 by sybalsky:mv:envos
```

;; Format a document for hardcopy. Returns the number of pages printed (not the final page number!). Returns NIL if the before-print-fn said not to print.

;; TEXTSTREAM is either already a textstream or somehow denotes a tedit-formatted file, otherwise an error. We don't here try to decide that a non-formatted file is a plain text file, as opposed binary or anything else.

```
(RESETLST
  (SETQ TEXTSTREAM (if (TEXTSTREAM TEXTSTREAM T)
    elseif (TEDIT.FORMATTEDFILEP TEXTSTREAM)
    then (CL:UNLESS (\GETSTREAM TEXTSTREAM 'INPUT T)
      [RESETSAVE (SETQ TEXTSTREAM (OPENTEXTSTREAM TEXTSTREAM))
        \ (PROGN (CLOSE? OLDVALUE])
```

```

TEXTSTREAM
else (ERROR TEXTSTREAM "is not a Tedit stream"))
(PROG ((TEXTOBJ (FGETTSTR TEXTSTREAM TEXTOBJ))
[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 PAGERECTION 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))
[RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ "Hardcopy")
' (PROGN (\TEDIT.MARKINACTIVE OLDVALUE)
(SETQ PAGERECTION (FGETTOBJ TEXTOBJ TXTPAGEFRAMES))
(SETPFS FORMATTINGSTATE PRESSREGION TEDIT.DEFAULTPAGERECTION)
; 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
(CL:UNLESS QUIET (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:
(\TEDIT.FORMATBOX TEXTOBJ PRSTREAM (GETPFS FORMATTINGSTATE CHNO)
PAGERECTION 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 PAGERECTION (GETPFS FORMATTINGSTATE NEWPAGELAYOUT))
;; 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)))
(SETQ TARGETFILENAME (STREAMPROP PRSTREAM 'PDFTARGETINFO))
(CL:UNLESS WASOPEN
(SETQ PRSTREAM (CLOSEF PRSTREAM))
; Only if we created the image stream should we close it.
(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))
(CL:UNLESS QUIET
(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)))]
)

```

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

(DEFINEQ

(\TEDIT.FORMATBOX

```

[LAMBDA (TEXTOBJ PRSTREAM STARTINGCHNO PAGEREGION FORMATTINGSTATE)
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:10 by rmk
; Edited 21-Oct-2024 00:33 by rmk
; 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
(CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
; A normal text region. Fill it with text formatted the usual way.
; Only format if we're not looking for something else.
(TEDIT.SETQS (LINES NIL LAST-CHNO)
(\TEDIT.FORMATTEXTBOX TEXTOBJ PRSTREAM STARTINGCHNO PAGEREGION FORMATTINGSTATE)))
)
(FOLIO
(CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
; A Page Number. Fill it in according to the instructions
; Only format if we're not looking for something else.
(SETQ LINES (\TEDIT.FORMATFOLIO TEXTOBJ PRSTREAM FORMATTINGSTATE PAGEREGION)))
(HEADING
(CL:WHEN (\TEDIT.FORMAT.FOUNDBOX? PAGEREGION FORMATTINGSTATE)
; A Page heading. Fill it in from a text source we saved for the
; occasion.
; 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
  )
  )
  (SELECTION ) ; Do one or another box, depending on some criterion.
  (\TEDIT.THELP) ; 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)
  (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 9-Jan-2025 22:27 by rmk
- ; Edited 3-Jan-2025 14:29 by rmk
- ; Edited 24-Nov-2024 11:46 by rmk
- ; Edited 20-Nov-2024 12:37 by rmk
- ; Edited 17-Nov-2024 19:10 by rmk
- ; Edited 26-Oct-2024 10:43 by rmk
- ; Edited 13-Jun-2024 17:14 by rmk
- ; 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 SELPIECES from the FORMATTINGSTATE and use them to fill REGION on a page. Return a list of line descriptors which fill the region. The SELPIECES are constructed by \TEDIT.HARDCOPY.PAGEHEADINGS

```
(LET ((REGION (SCALEREGION (DSPSCALE NIL PRSTREAM)
                           (fetch (PAGEREGION REGIONSPEC) of PAGEREGION)))
      (LOCALINFO (fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION))
      (PAGE# (GETPFS FORMATTINGSTATE PAGE#))
      HEADINGTEXTOBJ HEADINGSTREAM HEADING)
  (DECLARE (SPECVARS PAGE#))
  (CL:WHEN [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 (GETSPC HEADING
                                                                              SPFIRST])
  (SETQ HEADINGTEXTOBJ (GETTSTR HEADINGSTREAM TEXTOBJ))
```

:: Insert the heading pieces into HEADINGTEXTOBJ

```
(\TEDIT.INSERTPIECES (GETSPC HEADING SPFIRST)
  (\TEDIT.ALIGNEDPIECE 1 HEADINGTEXTOBJ)
  HEADINGTEXTOBJ)
```

::

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

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

:: Format the next line from HEADINGTEXTOBJ pieces

```
(SETQ LINE (\TEDIT.FORMATLINE HEADINGSTREAM CHNO NIL REGION PRSTREAM FORMATTINGSTATE))
(SETQ FORCENEXTPAGE (EQ (CHARCODE FORM)
                       (FGETLD LINE FORCED-END)))
[SETQ YBOT (if YBOT
              then ; Take account of this line's height
                  (IDIFFERENCE YBOT (FGETLD LINE LHEIGHT))
              else ; First line: position it at the top of the region.
                  (IDIFFERENCE BOTTOM (FGETLD LINE LDESCENT])
  (SETYBOT LINE YBOT)
  (SETQ CHNO (FGETLD LINE LCHARLIM)) ; Set the start of the next line
  LINE))]
```

(\TEDIT.FORMATPAGE

```
[LAMBDA (TEXTOBJ PRSTREAM CHNO PAGEREGION FORMATTINGSTATE)
; Edited 19-Feb-2025 13:32 by rmk
; Edited 8-Feb-2025 21:13 by rmk
; Edited 11-Dec-2024 22:39 by rmk
; 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.

```
(CL:UNLESS (ILEQ CHNO TEXTLEN)
  (RETURN))
(SETQ PC (\TEDIT.ALIGNEDPIECE CHNO TEXTOBJ))
(SETQ NEWPARALOOKS (\TEDIT.APPLY.PARASTYLES (PPARALOOKS PC)
                                             PC TEXTOBJ)) ; Print in the usual region on the page
; RMK: Why both 'NEWPAGE LAYOUT and
; :NEW-PAGE-LAYOUT ?
```

```
(CL:WHEN (EQ 'NEWPAGE LAYOUT (GETPLOOKS NEWPARALOOKS FMTPARATYPE))
  ;; 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 (GETPLOOKS
      NEWPARALOOKS
      FMTUSERINFO)
      '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
  (DSPUSHSTATE PRSTREAM)
  (DSPROTATE 90 PRSTREAM)
  (DSPTRANSLATE 0 (IMINUS (ffetch (REGION HEIGHT) of REGION))
    PRSTREAM)
  (DSPCLIPPINGREGION REGION PRSTREAM) ; Clip to the whole sheet.
  (DSPRIGHTMARGIN (fetch (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 19-Feb-2025 13:32 by rmk
  ; Edited 8-Feb-2025 23:39 by rmk
  ; Edited 11-Dec-2024 22:37 by rmk
```

```

; Edited 24-Nov-2024 11:46 by rmk
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:16 by rmk
; Edited 26-Oct-2024 10:46 by rmk
; Edited 21-Oct-2024 00:33 by rmk
; Edited 13-Jun-2024 17:15 by rmk
; 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 PARALOOKS LHEIGHT PREVLINELINE 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 (FGETTOBJ TEXTOBJ STREAMHINT)
                                  CHNO LINE REGION PRSTREAM FORMATTINGSTATE))

```

```

    (SETQ FORCENEXTPAGE (AND (EQ (CHARCODE FORM)
                                 (FGETLD LINE FORCED-END))
                             'USERBREAK))

```

```

    (SETQ LHEIGHT (FGETLD LINE LHEIGHT))
    (SETQ PARALOOKS (FGETLD LINE LPARALOOKS))
    (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 (FGETLD LINE LCHARLIM))

```

```

        LINE)

```

```

        ((LISTGET (FGETPLOOKS PARALOOKS FMTUSERINFO)
                  'FOOTNOTE)

```

```

          ;; This paragraph is a footnote para.

```

```

          (CL:WHEN FORCENEXTPAGE ; HELP in original code. SHOULDNT ?
            (\TEDIT.THELP))

```

```

          (SETQ FOOTNOTELINES (\TEDIT.FORMAT.FOOTNOTE TEXTOBJ PRSTREAM LINE REGION
            FORMATTINGSTATE))

```

```

          (SETQ CHNO (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 (FGETLD (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 (FGETPLOOKS PARALOOKS FMTSPECIALY)
        (NOT (ZEROP (FGETPLOOKS PARALOOKS FMTSPECIALY)))
        (FGETLD LINE 1STLN))
    ; There is a special Y location for this paragraph. Move there
    (SETQ SPECIALYPOS (FGETPLOOKS PARALOOKS FMTSPECIALY)))
  ((AND COLUMN-YBASE (FGETLD LINE 1STLN)
        (EQ (FGETPLOOKS PARALOOKS FMTCOLUMN)
            'NEXT))
    ; This is the first line of a new column; back YBOT back down to match the prior column.
    (- COLUMN-YBASE (FGETLD LINE LDESCENT)))
  [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 (FGETPLOOKS PARALOOKS FMTBASETOBASE)
  [SETQ LHEIGHT (IPLUS (FGETLD LINE LDESCENT)
                      (FGETPLOOKS PARALOOKS FMTBASETOBASE)
                      (COND
                        ((FGETLD LINE 1STLN)
                         (IPLUS (FGETPLOOKS PARALOOKS
                                LEADBEFORE)
                                (FGETPLOOKS (GETLD
                                             PREVLIN
                                             LPARALOOKS
                                             )
                                             LEADAFTER)))
                        (T 0])
                      (COND
                        ((\FIRST-COLUMN-START LINE PARALOOKS)
                         (IDIFFERENCE (IMIN PRIOR-COLUMN-YBOT YBOT)
                                       LHEIGHT))
                        (T (IDIFFERENCE YBOT LHEIGHT)
                           ; Just starting out; find the line's position with respect to the top
                           ; of the region to be filled.
                           (IDIFFERENCE RTOP (FGETLD LINE LTRUEHEIGHT))
                           (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 (FGETPLOOKS (FGETLD LINE LPARALOOKS)
                                                                    FMTNEWPAGEBEFORE)
                                                                    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 (FGETPLOOKS PARALOOKS FMTNEWPAGEAFTER))
                 ; 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))
        (SETYBOT LINE YBOT)
        (CL:WHEN (\FIRST-COLUMN-START LINE PARALOOKS)
                 ;; 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 (FGETLD 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 9-Jan-2025 21:52 by rmk
; Edited 3-Jan-2025 14:28 by rmk
; Edited 24-Nov-2024 11:46 by rmk
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:16 by rmk
; Edited 26-Oct-2024 10:46 by rmk
; Edited 13-Jun-2024 17:15 by rmk
; Edited 10-May-2024 12:32 by rmk
; 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 PAGE# FOLIOFORMAT PRETEXT POSTTEXT INFOLIST)
      (DECLARE (SPECVARS PAGE#))
      (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#)
                                                T)))
                 ;; 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]
                 (TEDIT.INSERT FOLIOSTREAM (CONCAT PRETEXT PAGE# POSTTEXT)
                                1 NIL T)
                 (bind LINE YBOT FORCENEXTPAGE (TEXTLEN _ (TEXTLEN (GETTSTR FOLIOSTREAM TEXTOBJ)))
                      (BOTTOM _ (fetch (REGION BOTTOM) of REGION))
                      (CHNO _ 1) while (ILEQ CHNO TEXTLEN) until FORCENEXTPAGE
                 collect (SETQ LINE (\TEDIT.FORMATLINE FOLIOSTREAM CHNO NIL REGION PRSTREAM FORMATTINGSTATE))
                        (SETQ FORCENEXTPAGE (EQ (CHARCODE FORM)

```

```

(FGETLD LINE FORCED-END))
; Format the next possible line
[SETQ YBOT (if YBOT
             then
             ; Take account of this line's height
             (IDIFFERENCE YBOT (FGETLD LINE LHEIGHT))
             else
             ; First line: position it at the top of the region.
             (IDIFFERENCE BOTTOM (FGETLD LINE LDESCENT))
             (SETYBOT LINE YBOT)
             ; This line is still good
             (SETQ CHNO (FGETLD LINE LCHARLIM))
             ; Set the start of the next line
             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)
(FORMATTING T)
; we're just munching along formatting. Keep going.
(SEARCHING T)
; 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))

```

(\TEDIT.SKIP.SPECIALCOND)

```

[LAMBDA (TSTREAM LINE PARALOOKS CHNO)
; Edited 19-Feb-2025 13:32 by rmk
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:35 by rmk
; Edited 26-Oct-2024 10:27 by rmk
; 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)
(FSETLD LINE 1STLN T)
(FSETLD LINE LSTLN T)
(FSETLD LINE LHEIGHT 0)
(FSETLD LINE LASCENT 0)
(FSETLD LINE LDESCENT 0)
(FSETLD LINE LTRUEASCENT 0)
(FSETLD LINE LTRUEDESCENT 0)
(FSETLD LINE LCHARLIM (IPLUS CHNO (for PC (HEADINGTYPE _ (GETPLOOKS PARALOOKS FMPARASUBTYPE))
inpieces
(fetch (TEXTSTREAM PIECE) of TSTREAM)
while (AND (EQ 'PAGEHEADING (GETPLOOKS (PPARALOOKS PC)
FMPARATYPE))
(EQ HEADINGTYPE (GETPLOOKS (PPARALOOKS PC)
FMPARASUBTYPE))))
sum (PLEN PC]))
)

```

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

```
(DEFINEQ
```

(\TEDIT.HARDCOPY.PAGEHEADINGS)

```

[LAMBDA (TEXTOBJ CHNO FORMATTINGSTATE)
; Edited 19-Feb-2025 13:32 by rmk
; Edited 12-Jan-2025 17:31 by rmk
; Edited 10-Jan-2025 15:42 by rmk
; Edited 21-Oct-2024 00:33 by rmk
; 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 in TEXTOBJ, copying the pieces of the different heading types into SELPIECES in
:: FORMATTINGSTATE, and returning the starting CHNO of the first non-heading piece.

```
(CL:UNLESS FORMATTINGSTATE ; If it isn't there, we would lose the headings
(\TEDIT.HELP "NIL FORMATTINGSTATE"))
(bind HEADINGSUBTYPE (PC _ (\TEDIT.CHTOPC CHNO TEXTOBJ)) while (AND PC (EQ 'PAGEHEADING (GETPLOOKS
(PPARALOOKS PC)
(FMTPARATYPE)))
do (SETQ HEADINGSUBTYPE (GETPLOOKS (PPARALOOKS PC)
(FMTPARASUBTYPE)))
(for P (START _ CHNO)
inpieces PC while (AND (EQ 'PAGEHEADING (GETPLOOKS (PPARALOOKS P)
(FMTPARATYPE)))
(EQ HEADINGSUBTYPE (GETPLOOKS (PPARALOOKS P)
(FMTPARASUBTYPE))))
do ; We loop at least once, because P=PC satisfies the while. We need the CHNO, not the piece for the selpieces
(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)))
```

:: For backward compatibility, this uses the information in the pageformat to create SELPIECES covering the pretext, pageno, and posttest, where
:: the pageno is produced by the PAGENO image object. We create a scratch textstream so that we can use the standard TEDIT.INSERT and
:: TEDIT.INSERT.OBJECT, then throw it away. This only happens once, when this heading is encountered, even if the pieces are rendered on
:: multiple pages.

```
[LET ((FOLIOINFO (fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION))
INFOLIST FOLIOSTREAM FOLIOTEXTOBJ)
; Have to set the SPECIALX and SPECIALY according to the PX and PY. And PQUAD
(CL:WHEN FOLIOINFO
(SETQ INFOLIST (LISTGET FOLIOINFO 'FORMATINFO))
[SETQ FOLIOSTREAM (OPENTEXTSTREAM NIL NIL '(PARALOOKS , (LISTGET FOLIOINFO 'PARALOOKS)
LOOKS
, (LISTGET FOLIOINFO 'CHARLOOKS)]
(SETQ FOLIOTEXTOBJ (GETTSTR FOLIOSTREAM TEXTOBJ))
(CL:WHEN (CADR INFOLIST)
(TEDIT.INSERT FOLIOSTREAM (MKSTRING (CADR INFOLIST))))
(TEDIT.INSERT.OBJECT (TEDIT.PAGENO.CREATE (CAR INFOLIST))
FOLIOSTREAM)
(CL:WHEN (CADDR INFOLIST)
(TEDIT.INSERT FOLIOSTREAM (MKSTRING (CADDR INFOLIST))))
(LISTPUT (fetch PAGEHEADINGS of FORMATTINGSTATE)
'\TEDIT.PAGENO
(\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES 1 (TEXTLEN FOLIOTEXTOBJ)
FOLIOTEXTOBJ)))]
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-Feb-2025 13:32 by rmk
; Edited 8-Feb-2025 23:39 by rmk
; Edited 11-Dec-2024 20:52 by rmk
; Edited 24-Nov-2024 11:46 by rmk
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:22 by rmk
; Edited 26-Oct-2024 10:46 by rmk
; 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
              (GETLD LINE LCHARLIM]
              (GETLD LINE LCHARLIM]
              (SETQ LASTLINE (CAR (FLAST LINES))) ; Find the last line in this box (column or page)
              (CL:WHEN (AND ORPHAN (GETLD ORPHAN LSTLN)
              (NOT (GETLD ORPHAN 1STLN)))
              ;; 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 LCHARLAST)
              (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 (CL:IF ORPHAN
              (GETLD ORPHAN LCHAR1)
              (GETLD (CAR (FLAST ORIGINAL-LINES))
              LCHARLIM)))
              ([AND (NEQ FORCENEXTPAGE 'USERBREAK)
              (ILEQ CHNO (TEXTLEN TEXTOBJ))
              (OR (GETPLOOKS (GETLD LASTLINE LPARALOOKS)
              FMTHEADINGKEEP)
              (AND (GETPLOOKS (GETLD LASTLINE LPARALOOKS)
              FMTKEEP)
              (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 (GETPLOOKS (GETLD LASTLINE LPARALOOKS)
              FMTHEADINGKEEP)
              (AND (GETPLOOKS (GETLD LASTLINE LPARALOOKS)
              FMTKEEP)
              (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 (GETPLOOKS (GETLD LASTLINE LPARALOOKS)
              FMTHEADINGKEEP)
              (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 bottom of the column.
              (bind (YBOT _ (IDIFFERENCE (fetch (REGION BOTTOM) of REGION)
              (GETLD (CAR (FLAST FOOTNOTELINES))
              LDESCENT)))
              for LINE in (REVERSE FOOTNOTELINES) do (SETYBOT 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
              (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 (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)
(SETQ OTHEREDGE (\TEDIT.PAPERWIDTH PAPERSIZE LANDSCAPE?))
(SETQ TVAL (IPLUS OTHEREDGE TVAL])
(RETURN TVAL])

]; He specified this value as an offset from the opposite edge.
; Convert it.

(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)
(SETQ OTHEREDGE (\TEDIT.PAPERHEIGHT PAPERSIZE LANDSCAPE?))
(SETQ TVAL (IPLUS OTHEREDGE TVAL])
(RETURN TVAL])

]; He specified this value as an offset from the opposite edge.
; Convert it.

(\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 ; letter size paper, 8.5inx11in
((NIL Letter LETTER 8.5x11)
612)
((Legal LEGAL 8.5x14)
612)
((A4 a4) ; A4 ISO-size paper, 210mmx297mm
595)
(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
((IGEQL NUMBER 1000)
(push CHARS 'm)
(add NUMBER -1000))
((IGEQL NUMBER 900)
(push CHARS 'c)
(push CHARS 'm)
(add NUMBER -900))
((IGEQL NUMBER 500)
(push CHARS 'd)
(add NUMBER -500))
((IGEQL NUMBER 400)
(push CHARS 'c)
(push CHARS 'd)
(add NUMBER -400))
((IGEQL NUMBER 100)
(push CHARS 'c)
(add NUMBER -100))
((IGEQL NUMBER 90)
(push CHARS 'x)
(push CHARS 'c)
(add NUMBER -90))
((IGEQL NUMBER 50)
(push CHARS 'l)
(add NUMBER -50))
((IGEQL NUMBER 40)
(push CHARS 'x)
(push CHARS 'l)
(add NUMBER -40))
((IGEQL NUMBER 10)
(push CHARS 'x)
(add NUMBER -10))
((IGEQL NUMBER 9)
(push CHARS 'i)
(push CHARS 'x)
(add NUMBER -9))
((IGEQL NUMBER 5)
(push CHARS 'v)
(add NUMBER -5))
)
)

```

```

((IGE0 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] (* The caller wants his roman numerals upper case)
 (T (CONCATLIST (REVERSE CHARS]))
)

```

;; Page number image obj

(DEFINEQ

(TEDIT.PAGENO.CREATE

```

[LAMBDA (FORMAT) ; Edited 7-Jan-2025 14:14 by rmk
 ; Edited 3-Jan-2025 14:44 by rmk
(LET ((OBJ (IMAGEOBJCREATE NIL TEDIT.PAGENOOBJ.IMAGEFNS))
 (IMAGEOBJPROP OBJ 'FORMAT (OR FORMAT 'ARABIC))
 OBJ))

```

(\TEDIT.PAGENO.OBJINIT

```

[LAMBDA NIL ; Edited 7-Jan-2025 22:54 by rmk
 ; Edited 3-Jan-2025 15:01 by rmk
 (* jds "9-Feb-86 15:17")

```

;; Initialize the IMAGEFNS for a page-number image object

```

(DECLARE (GLOBALVARS TEDIT.PAGENOOBJ.IMAGEFNS))
(SETQ TEDIT.PAGENOOBJ.IMAGEFNS (IMAGEFNSCREATE (FUNCTION \TEDIT.PAGENO.DISPLAYFN)
 (FUNCTION \TEDIT.PAGENO.IMAGEBOXFN)
 (FUNCTION \TEDIT.PAGENO.PUTBOXFN)
 (FUNCTION \TEDIT.PAGENO.GETFN)
 [FUNCTION (LAMBDA (OBJ)
 (create IMAGEOBJ copying OBJ)
 (FUNCTION \TEDIT.PAGENO.BUTTONEVENTINFN)
 'NIL
 'NIL
 'NIL
 'NIL
 'NIL
 'NIL NIL 'NIL 'PageNumber])

```

(\TEDIT.PAGENO.BUTTONEVENTINFN

```

[LAMBDA (IMAGEOBJ WINDOW SELECTION X Y SELWINDOW TEXTSTREAM BUTTON OPERATION)
 ; Edited 3-Jan-2025 14:32 by rmk
 ; Edited 14-Aug-93 19:44 by rmk:

```

;; Allow the user to change the page-number printed format.

;;; the user has pressed a button inside the bitmap object IMAGEOBJ. Bring up a menu of bitmap edit operations.

```

(CL:WHEN (AND (EQ BUTTON 'LEFT)
 (EQ OPERATION 'NORMAL))
 (LET (FORMAT)
 [SETQ FORMAT (MENU (create MENU
 ITEMS _ ' (Arabic 'ARABIC)
 ("Lower Roman" 'LOWERROMAN)
 (" Upper Roman" 'UPPERROMAN)
 (CL:WHEN [AND FORMAT (NEQ FORMAT (IMAGEOBJPROP IMAGEOBJ 'FORMAT]
 (IMAGEOBJPROP IMAGEOBJ 'FORMAT FORMAT)
 'CHANGED)))]))

```

(\TEDIT.PAGENO.IMAGEBOXFN

```

[LAMBDA (OBJ IMAGESTREAM) ; Edited 3-Jan-2025 14:30 by rmk
 ; Edited 26-Aug-2024 09:36 by rmk
 ; Edited 3-Aug-2024 13:10 by rmk
 ; Edited 19-Jul-2024 23:26 by rmk
 ; Edited 11-Oct-2022 22:51 by rmk
 ; Edited 4-Oct-2022 11:59 by rmk

```

;; Creates the box for a page number, a place holder on the display, otherwise the properly formatted number. Looks come from the font.

;; Create the box for a menu button containing LABEL in font FONT on STREAM (NIL means display).

```

(DECLARE (USEDFFREE PAGE#))
(LET ((FONT (DSPFONT NIL IMAGESTREAM))
 (FORMAT (IMAGEOBJPROP OBJ 'FORMAT))
 YSIZE XSIZE)
 (SETQ YSIZE (FONTPROP FONT 'HEIGHT))
 (SETQ XSIZE (STRINGWIDTH (if (DISPLAYSTREAMP IMAGESTREAM)
 then (CONCAT "[P#" (SELECTQ FORMAT
 (SELECTQ FORMAT
 (LOWERROMAN "x")
 (UPPERROMAN "X")

```

```

                                (MKSTRING "1"))
                                "]")
else (SELECTQ FORMAT
        (LOWERROMAN (ROMANNUMERALS PAGE#))
        (UPPERROMAN (ROMANNUMERALS PAGE# T))
        (MKSTRING PAGE#)))
    FONT))
(create IMAGEBOX
  XSIZE _ XSIZE
  YSIZE _ YSIZE
  YDESC _ 0
  XKERN _ 0)

```

(\TEDIT.PAGENO.DISPLAYFN

```

[LAMBDA (OBJ IMAGESTREAM IMAGESTREAMTYPE HOSTSTREAM)
; Edited 3-Jan-2025 14:30 by rmk
; Edited 26-Aug-2024 09:36 by rmk
; Edited 3-Aug-2024 13:10 by rmk
; Edited 19-Jul-2024 23:26 by rmk
; Edited 11-Oct-2022 22:51 by rmk
; Edited 4-Oct-2022 11:59 by rmk
(* jds "30-Aug-84 11:24")

```

:: Display the page number on IMAGESTREAM, a place holder for display, otherwise a formatted number. Looks come from the font.

```

(DECLARE (USEDFREE PAGE#))
(LET [(FORMAT (IMAGEOBJPROP OBJ 'FORMAT)
  (PRIN3 (if (DISPLAYSTREAMP IMAGESTREAM)
    then (CONCAT "[P#" (SELECTQ FORMAT
      (SELECTQ FORMAT
        (LOWERROMAN "x")
        (UPPERROMAN "X")
        (MKSTRING "1"))
      ])")
    else (SELECTQ FORMAT
      (LOWERROMAN (ROMANNUMERALS PAGE#))
      (UPPERROMAN (ROMANNUMERALS PAGE# T))
      (MKSTRING PAGE#)))
    IMAGESTREAM])

```

(\TEDIT.PAGENO.GETFN

```

[LAMBDA (FILESTREAM)
; Edited 3-Jan-2025 14:13 by rmk
(LET ((X (READ FILESTREAM (FIND-READTABLE "INTERLISP" T)))
  OBJ)
  (SETQ OBJ (IMAGEOBJCREATE (CAR X)
    PAGENOOBJ.IMAGEFNS))
  (replace (IMAGEOBJ IMAGEOBJPLIST) of OBJ with (CDR X)
    OBJ))

```

(\TEDIT.PAGENO.PUTFN

```

[LAMBDA (OBJ FILESTREAM)
; Edited 3-Jan-2025 15:01 by rmk
(PRINT (CONS (fetch (IMAGEOBJ OBJECTDATUM) of OBJ)
  (fetch (IMAGEOBJ IMAGEOBJPLIST) of OBJ))
  FILESTREAM
  (FIND-READTABLE "INTERLISP"]])
)

```

(\TEDIT.PAGENO.OBJINIT)

:: Foot note support

(DEFINEQ

(\TEDIT.FORMAT.FOOTNOTE

```

[LAMBDA (TEXTOBJ PRSTREAM LINE REGION FORMATTINGSTATE)
; Edited 20-Nov-2024 12:37 by rmk
; Edited 17-Nov-2024 19:22 by rmk
; Edited 13-Jun-2024 17:13 by rmk
; 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 (FGETTOBJ TEXTOBJ STREAMHINT)
  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
  (SETQ CHNO (FGETLD LINE LCHARLIM)) ; Keep track of the next character...
  (SETQ PREVLIN 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	20	\TEDIT.FORMAT.FOUNDBOX?	16	\TEDIT.PAGENO.IMAGEBOXFN	21
SCALEPAGEUNITS	19	\TEDIT.FORMATBOX	9	\TEDIT.PAGENO.OBJINIT	21
SCALEPAGEXUNITS	19	\TEDIT.FORMATFOLIO	15	\TEDIT.PAGENO.PUTFN	22
SCALEPAGEYUNITS	19	\TEDIT.FORMATHEADING	10	\TEDIT.PAPERHEIGHT	19
TEDIT.COMPOUND.PAGEFORMAT	5	\TEDIT.FORMATPAGE	11	\TEDIT.PAPERWIDTH	19
TEDIT.FORMAT.HARDCOPY	7	\TEDIT.FORMATTEXTBOX	12	\TEDIT.PARSE.PAGEFRAMES	3
TEDIT.GET.PAGEFORMAT	7	\TEDIT.HARDCOPY-COLUMN-END	17	\TEDIT.PUT.PAGEFRAMES	3
TEDIT.PAGEFORMAT	6	\TEDIT.HARDCOPY.PAGEHEADINGS	16	\TEDIT.SKIP.SPECIALCOND	16
TEDIT.PAGENO.CREATE	21	\TEDIT.PAGENO.BUTONEVENTINFN	21	\TEDIT.UNPARSE.PAGEFRAMES	3
TEDIT.SINGLE.PAGEFORMAT	4	\TEDIT.PAGENO.DISPLAYFN	22		
\TEDIT.FORMAT.FOOTNOTE	22	\TEDIT.PAGENO.GETFN	22		

VARIABLE INDEX

MAXPAGE#	3	TEDIT.DEFAULT.FOLIO.LOOKS	3	TEDIT.PAPER.SIZES	20
MINPAGE#	3	TEDIT.PAGE.FRAMES	7		

MACRO INDEX

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

RECORD INDEX

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