

File created: 19-Feb-2025 12:18:40 {WMEDLEY}<library>tedit>TEDIT-TFBRAVO.;175

edit by: rmk

changes to: (RECORDS PARA)
(FNS TEDITFROMBRAVO \TFBRAVO.READ.PARALOOKS \TFBRAVO.HANDLE.HEADING \TFBRAVO.PARSE.PROFILE.PARA
\TFBRAVO.SPLIT.PARA \TFBRAVO.RUN.TABSPEC \TFBRAVO.ADD.NAMEDTAB)

previous date: 8-Feb-2025 23:19:34 {WMEDLEY}<library>tedit>TEDIT-TFBRAVO.;174

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ TEDIT-TFBRAVOCOMS

```
[[DECLARE%: EVAL@COMPILE DONTCOPY ; Compile-time needs
  (FILES TEDIT-EXPORTS.ALL)
  (RECORDS BRAVOFONT PARA RUN)
  (MACROS \TFBRAVO.GETFONT \TOPOINTS)
  (CONSTANTS (BRAVO.TRAILER.CHARS ' (l d z x e y k j c q % ( % ) , s S u U b B i I g G v V w W t f o
    % \ 0 1 2 3 4 5 6 7 8 9 ]
```

:: Interface to TEDIT

```
(FNS TEDIT.BRAVOFILE? TEDITFROMBRAVO)
(ADDVARS (TEDIT.INPUT.FORMATS (TEDIT.BRAVOFILE? TEDITFROMBRAVO)))
```

:: Initial looks, USER.CM

```
(FNS \TFBRAVO.GET.USER.CM \TFBRAVO.USER.CM.LOOKS \TFBRAVO.READ.USER.CM \TFBRAVO.INIT.PARALOOKS
\TFBRAVO.INIT.PAGEFORMAT \TFBRAVO.GETPARAMS \TFBRAVO.FIND.LAST.TRAILER)
```

:: Decoding the Bravo file

```
(FNS \TFBRAVO.PARSE.PARA \TFBRAVO.READ.PARALOOKS \TFBRAVO.CREATE.RUNS \TFBRAVO.READ.CHARLOOKS
\TFBRAVO.FONT.FROM.CHARLOOKS \TFBRAVO.READNUM?)
```

:: Profile paragraphs

```
(FNS \TFBRAVO.HANDLE.HEADING \TFBRAVO.PARSE.PROFILE.PARA)
```

:: Creating the text stream

```
(FNS \TFBRAVO.INSERT.PARA \TFBRAVO.INSERT.RUN \TFBRAVO.SPLIT.PARA \TFBRAVO.RUN.TABSPEC
\TFBRAVO.INSTALL.PAGEFORMAT)
(FNS \TFBRAVO.ASSERT \TEST.CHARACTER.LOOKS \TEST.PARAGRAPH.LOOKS)
(INITVARS (TEDIT-DEFAULT-USER.CM "TEDIT-DEFAULT-USER.CM")
  (USER.CM.RDTBL (COPYREADTABLE))
  (PROFILE.PARA.RDTBL (COPYREADTABLE)))
```

```
(P (SETBRK (CHARCODE (% , %: = CR)
  NIL USER.CM.RDTBL)
  (SETSEPR ' (% )
  NIL USER.CM.RDTBL)
  (SETSYNTAX (CHARCODE %:)
  'SEPRCHAR PROFILE.PARA.RDTBL)
  (SETSYNTAX (CHARCODE CR)
  'BREAKCHAR PROFILE.PARA.RDTBL)
  (SETSYNTAX (CHARCODE ^Z)
  'BREAKCHAR PROFILE.PARA.RDTBL))
```

(COMS ; Named tabs. To be removed

```
(FNS \TFBRAVO.ADD.NAMEDTAB \TFBRAVO.COPY.NAMEDTAB \TFBRAVO.PUT.NAMEDTAB \TFBRAVO.GET.NAMEDTAB
\NAMEDTABNYET \NAMEDTABSIZE \NAMEDTABPREPRINT \TEDIT.NAMEDTAB.INIT)
(GLOBALVARS \NAMEDTAB.IMAGEFNS)
(DECLARE%: DONTVAL@LOAD DOCOPY (P (AND NIL (\TEDIT.NAMEDTAB.INIT])
```

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

```
(FILESLOAD TEDIT-EXPORTS.ALL)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(RECORD BRAVOFONT (BFFONTNUM BRFAMILY BRFSIZE BRWEIGHT BRFSLOPE))
```

```
[RECORD PARA (PARAFMTSPEC RUNS FORMATPTRS)
  (ACCESSFNS (PARATABDEFS (GETPLOOKS (fetch (PARA PARAFMTSPEC) of DATUM)
    FMTUSERINFO)
    (FSETPLOOKS (fetch (PARA PARAFMTSPEC) of DATUM)
    FMTUSERINFO NEWVALUE])
```

```
[RECORD RUN (RUNLENGTH RUNLOOKS RUNSTART RUNLAST)
  (ACCESSFNS (RUNTABS (fetch (CHARLOOKS CLUSERINFO) of (fetch (RUN RUNLOOKS) of DATUM))
    (replace (CHARLOOKS CLUSERINFO) of (fetch (RUN RUNLOOKS) of DATUM) with NEWVALUE])
)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS \TFBRAVO.GETFONT MACRO [(FONTNUM FIELD)
  (fetch (BRAVOFONT FIELD) of (FASSOC FONTNUM (FASSOC 'Font USER.CM.ALIST])
```

```
(PUTPROPS \TOPOINTS MACRO ((DIMENSION)
```

:: Assumes that the next token in LINE is a number to be converted to points, according to the conventions
:: specified in the Bravo user manual. Negative distances are relative to an 8.5 x 11 US Letter page.
:: This positions LINE at the token after the unit, if any.

```
(LET ((NUM (pop LINE))
      (UNIT (CAR LINE)))
  [SETQ UNIT (SELECTQ (U-CASE UNIT)
                    ((IN INCH INCHES %)
                     (pop LINE)
                     'INCH)
                    ((CM CMS)
                     (pop LINE)
                     'CM)
                    ((POINT POINTS PT PTS)
                     (pop LINE)
                     'POINT)
                    (CL:IF (FLOATP NUM)
                             'INCH
                             'POINT))]
  (SETQ NUM (SELECTQ UNIT
                    (INCH (FIXR (TIMES NUM 72)))
                    (CM (FIXR (FQUOTIENT (TIMES NUM 2.54 72))))
                    (NUM))
  (CL:WHEN (ILESSP NUM 0)
    (SETQ NUM (SELECTQ DIMENSION
                      (HEIGHT (IPLUS (CONSTANT (ITIMES 11 72))
                                       NUM))
                      (WIDTH (IPLUS (CONSTANT (FIX (FTIMES 8.5 72)))
                                       NUM))
                      (NIL NUM)
                      (\TEDIT.THELP "UNKNOWN DIMENSION" DIMENSION))))
  NUM))
)
```

(DECLARE%: EVAL@COMPILE

(RPAQQ BRAVO.TRAILER.CHARS

(l d z x e y k j c q (%) %, s S u U b B i I g G v V w W t f o % \ 0 1 2 3 4 5 6 7 8 9))

[CONSTANTS (BRAVO.TRAILER.CHARS ' (l d z x e y k j c q (%) %, s S u U b B i I g G v V w W t f o % \ 0 1 2 3 4
5 6 7 8 9)

)
)

:: Interface to TEDIT

(DEFINEQ

(TEDIT.BRAVOFILE?

[LAMBDA (STREAM TEXTOBJ)

; Edited 28-Nov-2023 10:34 by rmk
; Edited 17-Aug-2023 08:09 by rmk
; Edited 11-Aug-2023 22:59 by rmk
; Edited 5-Aug-2023 23:05 by rmk
; Edited 1-Aug-2023 08:15 by rmk
(* gbn " 3-Jun-85 21:06")

:: T if the open STREAM looks like a Bravo file.

```
(PROG (PLOOKS ENDCONDITION (ORIGINAL.FILE.POSITION (GETFILEPTR STREAM))
      NAME DIRS USER.CM) ; first look for a ^z, (beginning of a Bravo trailer)
  (CL:UNLESS (\TFBRAVO.FIND.LAST.TRAILER STREAM)
    (SETFILEPTR STREAM ORIGINAL.FILE.POSITION)
    (RETURN NIL)) ; BIN past the ^z
  (BIN STREAM)
  (SETQ PLOOKS (\TEST.PARAGRAPH.LOOKS STREAM)) ; if the next symbol is a slash then check if the character looks
; are valid
  [SETQ ENDCONDITION (CL:WHEN (EQ (CAR PLOOKS)
                                  '\/)
    (repeatuntil (\TEST.CHARACTER.LOOKS STREAM)))]
  (SETFILEPTR STREAM ORIGINAL.FILE.POSITION)
  (CL:WHEN (EQ ENDCONDITION 'BADLOOKS)
    (RETURN NIL))
  (RETURN T])
```

(TEDITFROMBRAVO

[LAMBDA (BFILE TEXTSTREAM PROPS USER.CM)

; Edited 19-Feb-2025 12:13 by rmk
; Edited 8-Feb-2025 23:03 by rmk
; Edited 2-Jan-2025 22:22 by rmk
; Edited 17-Jan-2024 12:11 by rmk
; Edited 26-Nov-2023 00:29 by rmk
; Edited 14-Nov-2023 17:09 by rmk
; Edited 22-Sep-2023 08:53 by rmk
; Edited 20-Aug-2023 20:25 by rmk
; Edited 18-Aug-2023 22:18 by rmk
; Edited 17-Aug-2023 10:17 by rmk
; Edited 13-Jun-90 01:00 by mitani

;;; Top level entry for conversion from a Bravo file to a textstream. The textstream is returned, "Writing" here means sticking it in the textstream, not saving to a Tedit file. Assumes that a stream BFILE is positioned at the first byte to be included.

```
(RESETLST
  (CL:UNLESS TEXTSTREAM
    (SETQ TEXTSTREAM (OPENTEXTSTREAM NIL)))
  ; Produce the USER.CM's alist of default values
  (bind PARA NEXTPARALOOKS USER.CM.CHARLOOKS USER.CM.PARALOOKS USER.CM.ALIST START (BSTREAM _ BFILE)
    (TEXTOBJ _ (TEXTOBJ TEXTSTREAM)) declare (SPECVARS USER.CM.PARALOOKS USER.CM.CHARLOOKS USER.CM.ALIST
      )
    first (CL:UNLESS (SETQ USER.CM (\TFBRAVO.GET.USER.CM BFILE USER.CM TEXTOBJ))
      ; Go for plain text
      (RETURN))
    (SETTOBJ TEXTOBJ FORMATTEDP T)
    (\TFBRAVO.USER.CM.LOOKS USER.CM TEXTOBJ) ; Set up the USER.CM look defaults
    (CL:UNLESS (GETSTREAM BSTREAM 'INPUT T) ; We keep it open, since we point to it
      (SETQ BSTREAM (OPENSTREAM BFILE 'INPUT)))
    (STREAMPROP BSTREAM :EXTERNAL-FORMAT :THROUGH)
    (PUTTEXTPROP TEXTOBJ 'OUTPUT-FORMAT :DEFAULT)
    [RESETSAVE (STREAMPROP BSTREAM 'ENDOFSTREAMOP (FUNCTION NIL))
      (PROGN (STREAMPROP ,BSTREAM 'ENDOFSTREAMOP OLDVALUE)]
    (SETQ NEXTPARALOOKS USER.CM.PARALOOKS)
    eachtime (SETQ START (GETFILEPTR BSTREAM)) ; Profiles and headings have to back up
      (SETQ PARA (\TFBRAVO.PARSE.PARA NEXTPARALOOKS BSTREAM TEXTOBJ))
      ;; No runs signals the very end
    while (fetch (PARA RUNS) of PARA) do (SETQ NEXTPARALOOKS (fetch (PARA PARAFMTSPEC) of PARA))
      ;; Valid profile paragraphs have a special interpretation, invalid ones must be mismarked
      ;; ordinary text
      (CL:UNLESS (AND (EQ 'PROFILE (GETPLOOKS NEXTPARALOOKS FMTPARATYPE)
        )
        (\TFBRAVO.PARSE.PROFILE.PARA BSTREAM PARA TEXTOBJ
          START))
        (\TFBRAVO.INSERT.PARA PARA BSTREAM TEXTOBJ))
    finally (\TFBRAVO.INSTALL.PAGEFORMAT TEXTOBJ)
      ;; Named tab information is collected in the userinfo fields, but then ignored.
      (for PARALOOKS in (GETTOBJ TEXTOBJ TXTPARALOOKSLIST) do (SETPLOOKS PARALOOKS FMTUSERINFO NIL))
      (for CHARLOOKS in (GETTOBJ TEXTOBJ TXTCHARLOOKSLIST) do (SETCLOOKS CHARLOOKS CLUSERINFO NIL))
      (\TEDIT.UNIQUIFY.ALL TEXTOBJ) ; Lists are complete and unique
      (\TEDIT.TRANSLATE.ASCIICHARS TEXTOBJ)
      (TEDIT.SETSEL TEXTOBJ 1 0 'LEFT)
      (RETURN TEXTSTREAM))))
)

(ADDTOVAR TEDIT.INPUT.FORMATS (TEDIT.BRAVOFILE? TEDITFROMBRAVO))

;; Initial looks, USER.CM

(DEFINEQ
  (\TFBRAVO.GET.USER.CM
    [LAMBDA (BFILE CANDIDATE TEXTOBJ)
      ; Edited 28-Nov-2023 17:38 by rmk
      ; Edited 11-Sep-2023 13:15 by rmk
      ; Edited 19-Aug-2023 23:24 by rmk
      ; Edited 17-Aug-2023 09:46 by rmk

      ;; BFILE's directory, connected directory, logindirectory, DIRECTORIES
      (DECLARE (USEDFREE TEDIT-DEFAULT-USER.CM))
      (CL:WHEN (STREAMP BFILE)
        (SETQ BFILE (FULLNAME BFILE)))
      (CL:WHEN (EQ CANDIDATE T)
        ; Because the test function's non-NIL value is passed in as
        ; CANDIDATE.
        (SETQ CANDIDATE NIL))
      ;; Returns the name of the user.cm file to be used in the conversion of Bravo BFILE. If CANDIDATE can't be found, the heuristic search is to
      ;; search in the following order:
      (PROG [USER.CM (DIRS ` (, (PACKFILENAME 'NAME NIL 'EXTENSION NIL 'VERSION NIL 'BODY BFILE)
        T NIL ,@DIRECTORIES)
        (CL:WHEN [AND CANDIDATE (SETQ USER.CM (CL:IF (STREAMP CANDIDATE)
          (FULLNAME CANDIDATE)
          (FINDFILE CANDIDATE T DIRS)))]
          (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "USER.CM = " USER.CM)
            T)
          (RETURN USER.CM))
        [SETQ USER.CM (OR (FINDFILE 'USER.CM T DIRS)
          (AND TEDIT-DEFAULT-USER.CM (FINDFILE TEDIT-DEFAULT-USER.CM T DIRS)
            (* (SELECTQ MKATOM (U-CASE
              (TEDIT.GETINPUT TEXTOBJ (CONCAT "USER.CM = "
                USER.CM " ? ")))) (NIL (TEDIT.PROMPTPRINT TEXTOBJ
                  "Yes" T) ((Y YES T) T) NIL))
            (if USER.CM
              then (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "USER.CM = " USER.CM)
                T)
              else do (SETQ USER.CM (TEDIT.GETINPUT TEXTOBJ "USER.CM file: (CR suppresses BRAVO conversion) "))
                (CL:WHEN (OR (NULL USER.CM)

```

```

(INFILEP USER.CM))
(RETURN))
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT USER.CM " not found")
T T)
(DISSMISS 3000)))
(RETURN USER.CM])

```

(\TFBRAVO.USER.CM.LOOKS

```

[LAMBDA (USER.CM TEXTOBJ) ; Edited 8-Feb-2025 22:13 by rmk
; Edited 2-Jan-2025 11:06 by rmk
; Edited 18-Aug-2023 18:47 by rmk
; Edited 16-Aug-2023 21:33 by rmk
; Edited 5-Aug-2023 17:15 by rmk
(DECLARE (USEDFREE USER.CM.CHARLOOKS USER.CM.PARALOOKS USER.CM.ALIST))
(SETQ USER.CM.ALIST (\TFBRAVO.READ.USER.CM USER.CM))
(SETQ USER.CM.CHARLOOKS (create CHARLOOKS
CLOFFSET _ 0))
(\TFBRAVO.FONT.FROM.CHARLOOKS USER.CM.CHARLOOKS (\TFBRAVO.GETFONT 0 BRFAMILY)
(\TFBRAVO.GETFONT 0 BRFSIZE))
(\TFBRAVO.INIT.PAGEFORMAT TEXTOBJ)
(SETQ USER.CM.PARALOOKS (\TFBRAVO.INIT.PARALOOKS USER.CM.ALIST))
(SETQ USER.CM.CHARLOOKS (\TEDIT.UNIQIFY.CHARLOOKS USER.CM.CHARLOOKS TEXTOBJ))
(SETQ USER.CM.PARALOOKS (\TEDIT.UNIQIFY.PARALOOKS USER.CM.PARALOOKS TEXTOBJ))
(SETTOBJ TEXTOBJ DEFAULTCHARLOOKS USER.CM.CHARLOOKS)
(SETTOBJ TEXTOBJ DEFAULTPARALOOKS USER.CM.PARALOOKS])

```

(\TFBRAVO.READ.USER.CM

```

[LAMBDA (USER.CM) ; Edited 27-Aug-2024 18:12 by rmk
; Edited 18-Aug-2023 22:26 by rmk
; Edited 10-Aug-2023 13:02 by rmk
; Edited 7-Aug-2023 12:52 by rmk
; Edited 1-Aug-2023 22:11 by rmk
; Edited 30-Jul-2023 18:57 by rmk
(* gbn "17-Sep-84 18:53")
(CL:UNLESS USER.CM
(SETQ USER.CM 'USER.CM))

```

:: digests a user.cm file returning an alist of contents. Returns ((Font)) if no bravo section of user.cm file

```

(RESETLST
(PROG (ALIST LINE)
(CL:UNLESS (GETSTREAM USER.CM 'INPUT T)
[RETSAVE (SETQ USER.CM (OPENSTREAM USER.CM 'INPUT 'OLD))
'(PROGN (CLOSEF? OLDVALUE))]
(SETFILEINFO USER.CM 'EOL 'ANY)
(AND NIL (STREAMPROP USER.CM :EXTERNAL-FORMAT :THROUGH))
(CL:UNLESS (AND (FILEPOS "[BRAVO]" USER.CM NIL NIL NIL T)
(EQ (CHARCODE EOL)
(READCCODE USER.CM)))
(RETURN NIL))

```

:: Read lines of the user.cm file until getting the empty line caused by eof (and the errortypelst entry) or until a line starts with '[' .

```

LLP (CL:UNLESS (NLSETQ (SETQ LINE (RATOMS (CONSTANT (CHARACTER (CHARCODE EOL)))
USER.CM USER.CM.RDTBL)))
(CL:UNLESS (ASSOC 'DefaultTab ALIST)
(push ALIST (CONS 'DefaultTab DEFAULTTTAB)))
(RETURN ALIST)) ; If the '[BRAVO]' section is the last one
(COND
((NULL LINE) ; ignore blank lines
(GO LLP))
((EQ (CAR LINE)
'END.OF.FILE)
(RETURN ALIST))
((EQ (NTHCHAR (CAR LINE)
1)
'%[]))

```

:: if '[' is the first character of the line, return the alist so far, because this is the beginning of the next section of the user.cm

```

(RETURN ALIST))
((NEQ (CADR LINE)
'%:))
(GO LLP))

```

:: CDDR to skip the :

```

(SELECTQ (PROG1 (CAR LINE)
(SETQ LINE (CDDR LINE)))
(FONT (CL:WHEN (FIXP (CAR LINE))
(NCONC1 [OR (FASSOC 'Font ALIST)
(CAR (PUSH ALIST (CONS 'Font)
(create BRAVOFONT
BFFONTNUM _ (POP LINE)
BRFAMILY _ (POP LINE)
BRFSIZE _ (POP LINE))))))
(TABS (SETQ ALIST (NCONC (\TFBRAVO.GETPARAMS LINE '((DefaultTab standard tab width))
'MICATOPOINTS)
ALIST)))

```

```
(MARGINS (SETQ ALIST (NCONC (\TFBRAVO.GETPARAMS LINE ' ((FirstLineLeftMargin paragraph margin
                                                                )
                                                                (LeftMargin left margin)
                                                                (RightMargin right margin))
                                                                'MICATOHALFPICAPPOINTS)
                                                                ALIST)))
(LEAD (SETQ ALIST (NCONC [\TFBRAVO.GETPARAMS LINE ' ((ParagraphLeading paragraph leading)
                                                                (LineLeading line leading)
                                                                ALIST)))
      (NIL)
      (GO LLP))))]
```

(\TFBRAVO.INIT.PARALOOKS

[LAMBDA (ALIST)

```
; Edited 8-Feb-2025 22:09 by rmk
; Edited 4-Aug-2024 22:17 by rmk
; Edited 28-Jul-2024 21:36 by rmk
; Edited 13-Aug-2023 11:27 by rmk
; Edited 8-Aug-2023 23:51 by rmk
; Edited 7-Aug-2023 14:59 by rmk
; Edited 31-May-91 15:26 by jds
```

```
:: creates the default paragraph looks from the USER.CM. The numeric values are Bravo defaults as specified in the Bravo documentation. This
:: assumes that all mica values in the USER.CM have already been converted to points.
```

```
(LET ((INITPARALOOKS (create PARALOOKS using TEDIT.DEFAULT.FMTSPEC)))
      ;; Bravo User Manual says that default tab is 36, the Bravo file format document says 60. I'm going with 36.
      (with PARALOOKS INITPARALOOKS (SETQ LEFTMAR (OR (CADR (ASSOC 'LeftMargin ALIST))
                                                         85))
        (SETQ 1STLEFTMAR (OR (CADR (ASSOC 'FirstLineLeftMargin ALIST))
                             LEFTMAR))
        (SETQ RIGHTMAR (OR (CADR (ASSOC 'RightMargin ALIST))
                            527))
        (SETQ LINELEAD (OR (CADR (ASSOC 'LineLeading ALIST))
                            1))
        (SETQ LEADBEFORE (OR (CADR (ASSOC 'ParagraphLeading ALIST))
                              0))
        (SETQ LEADAFTER 0)
        (SETQ FMTDEFAULTTAB (OR (CADR (ASSOC 'DefaultTab ALIST))
                                 DEFAULTTAB))
        (SETQ FMTSPECIALX 0)
        (SETQ FMTSPECIALY 0))
      INITPARALOOKS])
```

(\TFBRAVO.INIT.PAGEFORMAT

[LAMBDA (TEXTOBJ)

```
; Edited 22-Sep-2023 20:03 by rmk
; Edited 10-Aug-2023 10:02 by rmk
(* gbn "31-May-85 17:13")
```

```
:: Page numbers centered and 1/2 inch from top of US Letter page. One inch top/bottom margins
```

```
(PUTTEXTPROP TEXTOBJ 'PAGENUMBERS T)
(PUTTEXTPROP TEXTOBJ 'PAGENUMBERX (FIXR (FQUOTIENT (TIMES 8.5 72)
                                                    2)))
(PUTTEXTPROP TEXTOBJ 'PAGENUMBERY (IDIFFERENCE (ITIMES 11 72)
                                                  36))
(PUTTEXTPROP TEXTOBJ 'TOPMARGIN 72)
(PUTTEXTPROP TEXTOBJ 'BOTTOMMARGIN 72)
(PUTTEXTPROP TEXTOBJ 'PAGENUMBER.NOTONFIRSTPAGE T])
```

(\TFBRAVO.GETPARAMS

[LAMBDA (LINE NAMES SCALE)

```
; Edited 10-Aug-2023 13:19 by rmk
; Edited 7-Aug-2023 12:34 by rmk
(* jds "27-Aug-84 09:37")
```

```
:: If SCALE is provided, the values after = are numbers that will be scaled by SCALE.
```

```
:: LINE is a list of tokens for a particular line in the USER.CM. It may contain several parameters separated by commas, where each parameter is
:: heading by a list of identifying names (CDR of the corresponding entry in NAMES) ending in =. (CAR N) for each N in NAMES is the tag that
:: identifies that parameter in the resulting alist.
```

```
(LET (SEGMENTS VALUE ALIST)
      ;; To simplify, first chop LINE into its comma-separated segments
      (for LTAIL PREV (START _ LINE) on LINE do (if (NULL (CDR LTAIL))
                                                      then (push SEGMENTS START)
                                                         ; last one
                                                      elseif (EQ '%, (CAR LTAIL))
                                                         then (CL:WHEN PREV
                                                             ; Cut off the comma
                                                             (RPLACD PREV NIL))
                                                         (push SEGMENTS START)
                                                         (SETQ START (CDR LTAIL))
                                                         (SETQ PREV START)
                                                      else (SETQ PREV LTAIL)))
      (SETQ SEGMENTS (DREVERSE SEGMENTS))
      [for N SEG in NAMES when [SETQ SEG (find S in SEGMENTS
                                                suchthat (for NTAIL on (CDR N) as SS in S
```

```

always (OR (AND (EQ '= SS)
              (NULL (CDR NTAIL)))
        (STRING.EQUAL SS (CAR NTAIL])

do (CL:WHEN (SETQ VALUE (CADR (MEMB '= SEG)))
    (PUSH ALIST (LIST (CAR N)
                     (SELECTQ SCALE
                        (MICATOPOINTS (FIXR (FQUOTIENT VALUE MICASPERPT)))
                        (MICATOHALFPICAPOINTS
                         [ITIMES 6 (FIXR (FQUOTIENT VALUE (FTIMES MICASPERPT 6)
                                         VALUE))]))
    (DREVERSE ALIST])

```

(\TFBRAVO.FIND.LAST.TRAILER

[LAMBDA (BSTREAM)

; Edited 1-Aug-2023 23:35 by rmk
; Edited 8-Sep-2022 17:15 by rmk
(* jds "27-Dec-84 19:13")

:: scans backwards from the end of the file trying to find the beginning of the last Bravo trailer. Returns NIL if not found, otherwise T

```

(LET [(STREAM (GETSTREAM BSTREAM 'INPUT)
      (SETFILEPTR STREAM -1)
      (CL:WHEN (AND (IGREATERP (GETFILEPTR STREAM)
                              0)
                    (EQ (\BACKBIN STREAM)
                        (CHARCODE CR)))
              (bind C while (AND (SETQ C (\BACKBIN STREAM))
                                (FMEMB (CHARACTER C)
                                       BRAVO.TRAILER.CHARS))
              do NIL)
      (EQ (\PEEKBIN STREAM)
          (CHARCODE ^Z)))]])

```

; empty files are not Bravo files. It says here!

)

:: Decoding the Bravo file

(DEFINEQ

(\TFBRAVO.PARSE.PARA

[LAMBDA (OLDPARALOOKS BSTREAM TEXTOBJ)

; Edited 8-Feb-2025 23:04 by rmk
; Edited 21-Oct-2024 00:33 by rmk
; Edited 14-Nov-2023 13:03 by rmk
; Edited 7-Nov-2023 21:53 by rmk
; Edited 21-Aug-2023 23:41 by rmk
; Edited 20-Aug-2023 22:48 by rmk
; Edited 16-Aug-2023 21:28 by rmk
; Edited 13-Jun-2021 09:46 by rmk

:: OLDPARALOOKS are the paragraph looks of the previous paragraph, and RUNi are the character runs in the form returned by
:: \TFBRAVO.READ.CHARLOOKS, except that here we fill in the character count for the last run. Leaves the input file pointer at the end of the
:: trailer, after the CR.

:: ^Z marks the end of a Bravo-looks paragraph which may have internal CR's that mark the end of Tedit paragraphs. The Bravo runs with different
:: charlooks want to end up in different pieces all within the same paragraph.

:: The carriage return that ends the trailer is its own final run, the trailer itself is skipped.

```

(DECLARE (USEDFREE USER.CM.CHARLOOKS USER.CM.PARALOOKS))
(LET (BYTE PLEN ^ZPTR ENDCHAR PARALOOKS RUNS FORMATPTRS PARAGRAPH TABPTRS (PSTART (GETFILEPTR BSTREAM)
    (PARALOOKS USER.CM.PARALOOKS))

```

:: BYTE=NIL at EOF, no terminating ^Z

```

(until (SELCHARQ (SETQ BYTE (BIN BSTREAM))
            (^Z
             (SETQ ^ZPTR (SUB1 (GETFILEPTR BSTREAM))) ; End of Bravo paragraph, maybe some looks
             (SETQ PLEN (IDIFFERENCE ^ZPTR PSTART)) ; Exclude the ^Z
             [AND NIL (CL:WHEN FORMATPTRS
                             (PUSH FORMATPTRS (CONS (CHARCODE ^Z)
                                                      (SUB1 ^ZPTR)))] ; Length of the Bravo paragraph without the ^Z
             (SETQ FORMATPTRS (DREVERSE FORMATPTRS))
             (SETQ TABPTRS (DREVERSE TABPTRS))
             T)

```

((CR FORM LF) ; Remember the position of an internal formatting char, i.e. the byte that perhaps should be the end of an
; internal paragraph.

```

[PUSH FORMATPTRS (CONS BYTE (SUB1 (GETFILEPTR BSTREAM)
NIL)
(TAB ; Collect tab byte positions
 (PUSH TABPTRS (SUB1 (GETFILEPTR BSTREAM)))
NIL)
(NIL T)
NIL))
(SELCHARQ BYTE
 (^Z (SETQ PARALOOKS (\TFBRAVO.READ.PARALOOKS OLDPARALOOKS BSTREAM TEXTOBJ)
 (SETQ RUNS (\TFBRAVO.CREATE.RUNS BSTREAM PSTART PLEN)))
(NIL)
(\TEDIT.THELP "Bravo paragraph not ending in ^Z, CR, EOF"))
(create PARA
 PARAFMTSPEC _ PARALOOKS

```

RUNS _ RUNS
FORMATPTRS _ FORMATPTRS])

(\TFBRAVO.READ.PARALOOKS

[LAMBDA (OLDPARALOOKS BSTREAM)

; Edited 19-Feb-2025 12:14 by rmk
; Edited 8-Feb-2025 23:04 by rmk
; Edited 19-Dec-2024 23:42 by rmk
; Edited 21-Oct-2024 00:27 by rmk
; Edited 27-Aug-2024 21:59 by rmk
; Edited 28-Jul-2024 21:39 by rmk
; Edited 9-Sep-2023 21:40 by rmk
; Edited 21-Aug-2023 21:43 by rmk
; Edited 20-Aug-2023 15:48 by rmk
; Edited 18-Aug-2023 23:08 by rmk
; Edited 15-Aug-2023 00:23 by rmk
; Edited 13-Aug-2023 19:58 by rmk
; Edited 3-Aug-2023 00:20 by rmk
; Edited 31-May-91 15:26 by jds

(DECLARE (USEDFREE USER.CM.PARALOOKS))

:: Decodes bravo paragraph looks into a TEDIT PARALOOKS. OLDPARALOOKS is used just for its tabs.

(PARALOOKS! OLDPARALOOKS)

(bind LMFLAG 1LMFLAG COMMAND TABX TABNAME NAMEDTABS (TABDEFAULT _ (GETPLOOKS USER.CM.PARALOOKS FMTDEFAULTTAB
)))

(NEWPARALOOKS _ (create PARALOOKS using USER.CM.PARALOOKS))

first (CL:UNLESS (EQ 'PROFILE (FGETPLOOKS OLDPARALOOKS FMTPARATYPE))

:: It appears that heading-tabs don't carry over to other paragraphs. Although maybe the default interval-tab does?

(SETQ TABDEFAULT (OR (FGETPLOOKS OLDPARALOOKS FMTDEFAULTTAB)
(FGETPLOOKS USER.CM.PARALOOKS FMTDEFAULTTTAB)))

:: We don't put the NAMEDTABS in the TABSPEC since we don't know which ones will be activated by any particular run.

(SETQ NAMEDTABS (COPY (FGETPLOOKS OLDPARALOOKS FMTUSERINFO)))

do (SELCHARQ (SETQ COMMAND (BIN BSTREAM))

(l (SETQ LMFLAG T)

(FSETPLOOKS NEWPARALOOKS LEFTMAR (\TFBRAVO.READNUM? BSTREAM T 'MICATOHALFPICAPOINTS)))

(d (SETQ 1LMFLAG T)

(FSETPLOOKS NEWPARALOOKS 1STLEFTMAR (\TFBRAVO.READNUM? BSTREAM T 'MICATOHALFPICAPOINTS)))

(z (FSETPLOOKS NEWPARALOOKS RIGHTMAR (\TFBRAVO.READNUM? BSTREAM T 'MICATOHALFPICAPOINTS)))

(x (FSETPLOOKS NEWPARALOOKS LINELEAD (\TFBRAVO.READNUM? BSTREAM T)))

(e (FSETPLOOKS NEWPARALOOKS LEADAFTER 0)

(FSETPLOOKS NEWPARALOOKS LEADBEFORE (\TFBRAVO.READNUM? BSTREAM T)))

(y ; vertical tabs are supported

(FSETPLOOKS NEWPARALOOKS FMTSPECIALX 0)

(FSETPLOOKS NEWPARALOOKS FMTSPECIALY (\TFBRAVO.READNUM? BSTREAM T)))

(k (FSETPLOOKS NEWPARALOOKS FMTHEADINGKEEP (\TFBRAVO.READNUM? BSTREAM T)))

(w 'HardcopyMode)

(j (FSETPLOOKS NEWPARALOOKS QUAD 'JUSTIFIED))

(c (FSETPLOOKS NEWPARALOOKS QUAD 'CENTERED))

(q :: Profiles are marked here but then interpreted at the top

(FSETPLOOKS NEWPARALOOKS FMTPARATYPE 'PROFILE))

(% (; Collect the named tabs

(SETQ TABX (\TFBRAVO.READNUM? BSTREAM T)) ; Name or X position

:: Tabs apparently round down/truncate, not up.

(SELCHARQ (SETQ COMMAND (BIN BSTREAM))

(% (SETQ TABDEFAULT (HCUNSCALE MICASPERPT TABX)))

(%, (CL:WHEN (IGREATERP TABX 14)

(\TEDIT.THELP TABX " is not a legal tab-name"))

(SETQ TABNAME (ADD1 TABX)) ; Adding 1 to align with t1, t2...

(SETQ TABX (\TFBRAVO.READNUM? BSTREAM T))

(CL:UNLESS (EQ (CHARCODE %))

(BIN BSTREAM))

(\TEDIT.THELP "MISSING CLOSING) IN TABSPEC"))

:: Here we collect the tabs declared in this paragraph or inherited from before. 65535 means delete that the
:: named tab (possibly inherited), otherwise the name is given a new TABX for all runs of this paragraph and
:: beyond.

[if (EQ TABX 65535)

then (SETQ NAMEDTABS (DREMOVE (ASSOC TABNAME NAMEDTABS)
NAMEDTABS))

else (RPLACD [OR (ASSOC TABNAME NAMEDTABS)

(CAR (push NAMEDTABS (CONS TABNAME)

(create TAB

TABX _ (HCUNSCALE MICASPERPT TABX)

TABKIND _ 'LEFT])

(\TEDIT.THELP "ILLFORMED BRAVO TAB SPEC")))

(SPACE)

(CR \)

(CL:WHEN (AND LMFLAG (NOT 1LMFLAG)) ; If there was a Left margin but no firstline left then default it

(FSETPLOOKS NEWPARALOOKS 1STLEFTMAR (FGETPLOOKS NEWPARALOOKS LEFTMAR)))

(FSETPLOOKS NEWPARALOOKS FMTDEFAULTTAB TABDEFAULT)

(FSETPLOOKS NEWPARALOOKS FMTUSERINFO (DREVERSE NAMEDTABS))

(CL:WHEN (EQ COMMAND (CHARCODE CR)) ; Read the \separator, but leave the terminating CR

(\BACKFILEPTR BSTREAM))

```
(RETURN NEWPARALOOKS))
(\TEDIT.THELP (CHARACTER COMMAND)
  " is not a legal Bravo paragraph-format character"])
```

(\TFBRAVO.CREATE.RUNS

```
[LAMBDA (BSTREAM PSTART PLEN) ; Edited 14-Nov-2023 13:01 by rmk
; Edited 9-Sep-2023 21:41 by rmk
; Edited 20-Aug-2023 23:03 by rmk

(bind RUNS RUN RLEN (RUNSTART _ PSTART)
(OLDCHARLOOKS _ USER.CM.CHARLOOKS) do (SETQ RUN (\TFBRAVO.READ.CHARLOOKS BSTREAM OLDCHARLOOKS
RUNSTART PLEN))

(push RUNS RUN)
(SETQ OLDCHARLOOKS (fetch (RUN RUNLOOKS) of RUN))
(CL:WHEN (fetch (RUN RUNLAST) of RUN)
  (RETURN (DREVERSE RUNS)))
(SETQ RLEN (fetch (RUN RUNLENGTH) of RUN))
; Set up for next run

(ADD RUNSTART RLEN)
(ADD PLEN (IMINUS RLEN))
(SETQ OLDCHARLOOKS (fetch (RUN RUNLOOKS) of RUN]))
```

(\TFBRAVO.READ.CHARLOOKS

```
[LAMBDA (BSTREAM OLDCHARLOOKS RUNSTART PLEN) ; Edited 2-Jan-2025 23:44 by rmk
; Edited 21-Oct-2024 00:27 by rmk
; Edited 9-Sep-2023 21:39 by rmk
; Edited 20-Aug-2023 16:15 by rmk
; Edited 18-Aug-2023 20:11 by rmk
; Edited 31-May-91 15:25 by jds
```

;; Read the character looks trailer building a TEDIT charlooks record. Most fields are immediately valid, however, the tabcolor is stored in the
;; cluserinfo field of the looks, and the font is still in numeric form
;; The charlooks trailer (from \ to CR) consists of a sequence of run-looks. Each run-look is a sequence of commands followed by the length of the
;; run. If the first run has no commands (i.e. the \ is followed immediately by a length-number), then the first run gets the USER.CM default looks.

```
(bind COMMAND LEN LAST VALUE TABNAMES FAMILY SIZE BOLD ITALIC (NEWCHARLOOKS _ (create CHARLOOKS using
OLDCHARLOOKS
)))
first [SETQ FAMILY (SETQ SIZE (SETQ BOLD (SETQ ITALIC 'OFF] until (SETQ LEN (\TFBRAVO.READNUM? BSTREAM))
do
;; Some command letters are followed by numeric arguments (f1 vs b). Any spaces around command letters are skipped. BIN is used here
;; for one-byte arguments, but perhaps a version that skips initial spaces would be safer?
(SELCHARQ (SETQ COMMAND (BIN BSTREAM))
  (s (FSETCLOOKS NEWCHARLOOKS CLSTRIKE T))
  (S (FSETCLOOKS NEWCHARLOOKS CLSTRIKE NIL))
  (u (FSETCLOOKS NEWCHARLOOKS CLULINE T))
  (U (FSETCLOOKS NEWCHARLOOKS CLULINE NIL))
  (b (SETQ BOLD T))
  (B (SETQ BOLD NIL))
  (i (SETQ ITALIC T))
  (I (SETQ ITALIC NIL))
  (g "Graphic T --unsupported")
  (G "Graphic NIL")
  (v (FSETCLOOKS NEWCHARLOOKS CLINVISIBLE NIL))
  (V (AND NIL (FSETCLOOKS NEWCHARLOOKS CLINVISIBLE T)))
  (t
;; Collect the named tabs for writerun
  (PUSH TABNAMES (CHARACTER (BIN BSTREAM))))
(f
; Save the fontface until the end
(SETQ VALUE (CHARACTER (BIN BSTREAM)))
(SETQ SIZE (\TFBRAVO.GETFONT VALUE BRSIZE))
(SETQ FAMILY (\TFBRAVO.GETFONT VALUE BRFAMILY)))
(o (SETQ VALUE (\TFBRAVO.READNUM? BSTREAM T)) ; Superscript
(FSETCLOOKS NEWCHARLOOKS CLOFFSET (CL:IF (IGREATERP VALUE 127)
(IDIFFERENCE VALUE 256)
VALUE)))
(SPACE)
(CR
;; We hit the trailer-terminating CR, It is either the end-marker for the last run, or a signal that this paragraph has no run-look
;; information.
(if (EQ 0 PLEN)
  then
;; If we have already accounted for all the pre-trailer characters, just return a trivial paragraph-final run pointing at
;; the end-of-trailer CR
  (SETQ RUNSTART (SUB1 (GETFILEPTR BSTREAM)))
  (SETQ LEN 1)
  (SETQ LAST T)
  else (\BACKFILEPTR BSTREAM) ; Leave the CR to be read on the next (PLEN=0) call.
; Otherwise, PLEN is what's left for the final substantive run
  (SETQ LEN PLEN))
(GO $$OUT))
(\TEDIT.THELP (CHARACTER COMMAND)
  " is not a legal Bravo command character look"))
finally
;; Wait til end to do font, so we have the bold/italic looks for sure. Last run may not have an explicit length
(FSETCLOOKS NEWCHARLOOKS CLUSERINFO (DREVERSE TABNAMES))
(\TFBRAVO.FONT.FROM.CHARLOOKS NEWCHARLOOKS FAMILY SIZE BOLD ITALIC)
(RETURN (create RUN
```



```

RUNSTART _ RUNSTART
RUNLENGTH _ LEN
RUNLOOKS _ NEWCHARLOOKS
RUNLAST _ LAST])

```

(\TFBRAVO.FONT.FROM.CHARLOOKS

```

[LAMBDA (CHARLOOKS FAMILY SIZE BOLD ITALIC) ; Edited 2-Jan-2025 23:43 by rmk
; Edited 1-Aug-2023 13:21 by rmk
; Edited 31-May-91 15:26 by jds

```

:: Takes a TEDIT CHARLOOKS with fields filled in (CLNAME = family name) and creates the font to fill it.

```

[LET ((OLDFONT (GETCLOOKS CHARLOOKS CLFONT)))
  (CL:WHEN (EQ FAMILY 'OFF)
    (SETQ FAMILY (FONTPROP OLDFONT 'FAMILY)))
  (CL:WHEN (EQ SIZE 'OFF)
    (SETQ SIZE (FONTPROP OLDFONT 'SIZE)))
  (CL:WHEN (EQ BOLD 'OFF)
    [SETQ BOLD (EQ 'BOLD (FONTPROP OLDFONT 'WEIGHT])]
  (CL:WHEN (EQ ITALIC 'OFF)
    [SETQ ITALIC (EQ 'ITALIC (FONTPROP OLDFONT 'SLOPE))]
  [SETCLOOKS CHARLOOKS CLFONT (FONTCREATE FAMILY SIZE (LIST (CL:IF BOLD
    'BOLD
    'MEDIUM)
    (CL:IF ITALIC
    'ITALIC
    'REGULAR)
    'REGULAR]
    (SETCLOOKS CHARLOOKS CLNAME (FONTUNPARSE (GETCLOOKS CHARLOOKS CLFONT)
CHARLOOKS]))

```

(\TFBRAVO.READNUM?

```

[LAMBDA (BSTREAM REQUIRED SCALE) ; Edited 10-Aug-2023 13:06 by rmk
; Edited 9-Aug-2023 07:53 by rmk
; Edited 5-Aug-2023 20:31 by rmk

```

:: If a digit appears as the first non-space character from the current stream position, the integer starting at that digit is returned. If MICAS, it is assumed that NUM is in micas and should be scaled to a multiple of 6 points (= 1/2 pica so that Tedit's margins aren't displayed as goofy :: decimals. The stream is left positioned before the first nondigit nonspace character.

```

(bind C (NUM _ 0) first (while (EQ (CHARCODE SPACE)
  (SETQ C (\PEEKCCODE BSTREAM T)))
  do (BIN BSTREAM)
  (CL:UNLESS (DIGITCHARP C)
  (CL:WHEN REQUIRED
  (ERROR "Bravo command without a numeric argument at position " (GETFILEPTR
  BSTREAM))))
  (RETURN NIL))
do [SETQ NUM (IPLUS (ITIMES NUM 10)
  (IDIFFERENCE (BIN BSTREAM)
  (CHARCODE 0)
repeatwhile (DIGITCHARP (\PEEKCCODE BSTREAM T))
finally (while (EQ (CHARCODE SPACE)
  (\PEEKCCODE BSTREAM T))
  do (BIN BSTREAM)
  (RETURN (SELECTQ SCALE
  (MICATOHALFPICAPPOINTS
  [ITIMES 6 (FIXR (FQUOTIENT NUM (FTIMES MICASPERPT 6]))
  (MICATOPPOINTS (FIXR (FQUOTIENT NUM MICASPERPT)))
  NUM])
)

```

:: Profile paragraphs

(DEFINEQ

(\TFBRAVO.HANDLE.HEADING

```

[LAMBDA (BSTREAM TEXTOBJ HEADINGSTART) ; Edited 19-Feb-2025 12:17 by rmk
; Edited 8-Feb-2025 23:05 by rmk
; Edited 20-Aug-2023 20:11 by rmk
; Edited 18-Aug-2023 10:37 by rmk
; Edited 12-Aug-2023 12:25 by rmk
; Edited 9-Aug-2023 23:37 by rmk
; Edited 4-Aug-2023 10:39 by rmk
; Edited 1-Aug-2023 22:24 by rmk
; Edited 31-May-91 15:26 by jds

```

:: Called from \TFBRAVO.PARSE.PROFILE.PARA. The heading is a paragraph beginning at the current position, presumably just a line with a :: looks trailer. Its paralooks have to be marked with special heading properties--heading type and special X and Y locations.

```

(DECLARE (USEDFREE USER.CM.PARALOOKS))
(LET (HEADINGDESC HEADINGPARA HEADINGPARALOOKS) ; skip over the trailer of the profile para
  (SETFILEPTR BSTREAM HEADINGSTART)
  (SETQ HEADINGPARA (\TFBRAVO.PARSE.PARA USER.CM.PARALOOKS BSTREAM TEXTOBJ))
  (SETQ HEADINGPARALOOKS (fetch (PARA PARAFMTSPEC) of HEADINGPARA))
  (SETPLOOKS HEADINGPARALOOKS FMTPARATYPE 'PAGEHEADING)

```

;; This is where the vertical tab info is placed for the heading, remove the special x and y and use them as the position for the descriptor

```
(SETQ HEADINGDESC (LIST (GENSYM 'PageHeading)
                        (OR (FGETPLOKS HEADINGPARALOOKS FMTSPECIALX)
                            0)
                        (OR (FGETPLOKS HEADINGPARALOOKS FMTSPECIALY)
                            0)))
(FSETPLOKS HEADINGPARALOOKS FMTPARASUBTYPE (CAR HEADINGDESC))
(FSETPLOKS HEADINGPARALOOKS FMTSPECIALX (CADR HEADINGDESC))
(FSETPLOKS HEADINGPARALOOKS FMTSPECIALY (CADDR HEADINGDESC))
; now write out the heading paragraph
(\TFBRAVO.INSERT.PARA HEADINGPARA BSTREAM TEXTOBJ MAX.FIXP)
HEADINGDESC])
```

(\TFBRAVO.PARSE.PROFILE.PARA

```
[LAMBDA (BSTREAM PARAGRAPH TEXTOBJ START)
```

```
; Edited 19-Feb-2025 12:17 by rmk
; Edited 8-Feb-2025 21:27 by rmk
; Edited 22-Sep-2023 20:02 by rmk
; Edited 19-Aug-2023 23:33 by rmk
; Edited 17-Aug-2023 14:51 by rmk
; Edited 10-Aug-2023 10:37 by rmk
; Edited 1-Aug-2023 13:29 by rmk
(* gbn " 3-Jun-85 17:23")
```

;; Parse a Bravo profile paragraph, and set up the corresponding TEdit page looks, headings, page numbers.

;; START is the beginning of the profile lines, current fileptr (END) is the end of its looks information (which is kind of odd, since "Margins:" has no looks. At best it sets carryover tabs.

;; This code processes the profile settings, storing them away as TEXTOBJ properties for later installation. Presumably, if there are multiple valid profile paragraphs in the same file, then the settings of the last one will override earlier ones.

;; Heading lines are followed by separate paragraphs containing the formatted text of the heading. The parameters are saved in the HEADINGDESC

;; Returns T if this is truly a profile paragraph, beginning with one of the profile keywords, otherwise NIL. Either way, BSTREAM is positioned at the beginning of the next unprocessed paragraph

;; START

```
(bind LINE ROMAN UPPERCASE (END _ (GETFILEPTR BSTREAM)) first (SETFILEPTR BSTREAM START)
while (ILESSP (GETFILEPTR BSTREAM)
      END)
```

do [.. Each RATOMS reads a line, the last line including the trailer characters. ^Z is a break so processing on the last line can stop at that token. Presumably the RATOMS will have reached END (and not gone past it), but we set the fileptr back to that position just in case, to set up for the next paragraph. Semi-colon is a sepr in the readable, we don't have to worry about those.

```
(SETQ LINE (U-CASE (RATOMS (CHARACTER (CHARCODE CR)
                                   BSTREAM PROFILE.PARA.RDTBL)))
```

;; Now we have to parse, checking and skipping the irrelevant keywords. Each keyword starts a line, each while runs until its end-token (CR?)

```
(SELECTQ (pop LINE)
(PAGE
```

; parse the page numbers stuff

```
(\TFBRAVO.ASSERT 'NUMBERS (pop LINE))
(while LINE do (SELECTQ (pop LINE)
  (NO (PUTTEXTPROP TEXTOBJ 'PAGENUMBERS 'NO))
  (YES ; this is default
    (PUTTEXTPROP TEXTOBJ 'PAGENUMBERS NIL))
  (FIRST (\TFBRAVO.ASSERT 'PAGE (pop LINE))
    (PUTTEXTPROP TEXTOBJ 'FIRSTPAGE NO (pop LINE)))
  (NOT-ON-FIRST-PAGE
    (PUTTEXTPROP TEXTOBJ 'PAGENUMBER.NOTONFIRSTPAGE T))
  (X (PUTTEXTPROP TEXTOBJ 'PAGENUMBERX (\TOPOINTS 'WIDTH)))
  (Y (PUTTEXTPROP TEXTOBJ 'PAGENUMBERY (\TOPOINTS 'HEIGHT)))
  (ROMAN (SETQ ROMAN T))
  (UPPERCASE (SETQ UPPERCASE T))
  NIL)))
```

(COLUMNS ; parse the columns numbers stuff

```
(PUTTEXTPROP TEXTOBJ 'NUMBEROFCOLUMNS (pop LINE))
(while LINE do (SELECTQ (pop LINE)
  (EDGE (\TFBRAVO.ASSERT 'MARGIN (pop LINE))
    (PUTTEXTPROP TEXTOBJ 'EDGEMARGIN (\TOPOINTS 'WIDTH)))
  (BETWEEN (\TFBRAVO.ASSERT 'COLUMNS (pop LINE))
    (PUTTEXTPROP TEXTOBJ 'BETWEENCOLUMNS (\TOPOINTS 'WIDTH)))
  NIL)))
```

(MARGINS ; parse the margins stuff

```
(while LINE do (SELECTQ (pop LINE)
  (TOP (PUTTEXTPROP TEXTOBJ 'TOPMARGIN (\TOPOINTS 'HEIGHT)))
  (BOTTOM (PUTTEXTPROP TEXTOBJ 'BOTTOMMARGIN (\TOPOINTS 'HEIGHT)))
  (BINDING (PUTTEXTPROP TEXTOBJ 'BINDING (\TOPOINTS)))
  NIL)))
```

(ODD (\TFBRAVO.ASSERT 'HEADING (pop LINE))

```
(CL:WHEN (EQ (CAR LINE)
             'NOT-ON-FIRST-PAGE)
```

```
(pop LINE)
(PUTTEXTPROP TEXTOBJ 'ODDHEADING.NOTONFIRSTPAGE T))
```

; Heading is on the next line

```
(PUTTEXTPROP TEXTOBJ 'ODDHEADINGDESC (\TFBRAVO.HANDLE.HEADING BSTREAM TEXTOBJ END))
```

;; We advance past the heading paragraph, presumably we are done.

```

        (SETQ END (GETFILEPTR BSTREAM))
    (EVEN (\TFBRAVO.ASSERT (pop LINE)
        'HEADING)
        (CL:WHEN (EQ (CAR LINE)
            'NOT-ON-FIRST-PAGE)
            (pop LINE)
            (PUTTEXTPROP TEXTOBJ 'EVENHEADING.NOTONFIRSTPAGE T))
        (PUTTEXTPROP TEXTOBJ 'EVENHEADINGDESC (\TFBRAVO.HANDLE.HEADING BSTREAM TEXTOBJ END))
        (SETQ END (GETFILEPTR BSTREAM)))
    (HEADING (CL:WHEN (EQ (CAR LINE)
        'NOT-ON-FIRST-PAGE)
            (pop LINE)
            (PUTTEXTPROP TEXTOBJ 'HEADING.NOTONFIRSTPAGE T))
        (PUTTEXTPROP TEXTOBJ 'HEADINGDESC (\TFBRAVO.HANDLE.HEADING BSTREAM TEXTOBJ END))
        (SETQ END (GETFILEPTR BSTREAM)))
    ((LINE PRIVATE) ; Line numbers, private data not supported
        NIL)
    (PROGN ; Not a profile line, presumably a mistaken q.
        (SETFILEPTR BSTREAM END)
        (FSETPLOOKS (fetch (PARA PARAFMTSPEC) of PARAGRAPH)
            FMTPARATYPE NIL)
        (RETURN NIL])
    repeatuntil [EQ (CAR LINE)
        (CONSTANT (CHARACTER (CHARCODE ^Z]
    finally (CL:WHEN ROMAN
        (PUTTEXTPROP TEXTOBJ 'ROMAN (CL:IF UPPERCASE
            'X
            'x)))
        ;; Set the file to the beginning of the next paragraph
        (SETFILEPTR BSTREAM END)
        (RETURN T])

```

;; Creating the text stream

(DEFINEQ

(\TFBRAVO.INSERT.PARA

```

[LAMBDA (PARA BSTREAM TEXTOBJ) ; Edited 8-Feb-2025 23:06 by rmk
; Edited 20-Aug-2023 16:13 by rmk

```

;; Inserts pieces into TEXTOBJ that correspond to the runs in PARA. PARA may be broken up at internal CR's to get spacing and tabs right.

```

(for P PARALOOKS in (\TFBRAVO.SPLIT.PARA PARA) do (SETQ PARALOOKS (fetch (PARA PARAFMTSPEC) of P))
    (for RUN in (fetch (PARA RUNS) of P)
        do (SETQ PARALOOKS (\TFBRAVO.RUN.TABSPEC RUN PARALOOKS)
            (\TFBRAVO.INSERT.RUN RUN BSTREAM PARALOOKS TEXTOBJ])

```

(\TFBRAVO.INSERT.RUN

```

[LAMBDA (RUN BSTREAM PARALOOKS TEXTOBJ) ; Edited 8-Feb-2025 23:08 by rmk
; Edited 17-Mar-2024 12:41 by rmk
; Edited 16-Jan-2024 18:28 by rmk
; Edited 29-Dec-2023 11:50 by rmk
; Edited 23-Sep-2023 12:11 by rmk
; Edited 23-Aug-2023 08:31 by rmk
; Edited 20-Aug-2023 16:12 by rmk
; Edited 19-Aug-2023 14:47 by rmk

```

;; A Bravo run can include many CR's each of which should end a separate TEDIT paragraph. Unless we want to think of those as paragraph
 ;; internal meta-CRs ?

;; PARALOOKS is the intended paragraph PARALOOKS for the paragraph, providing the margins, line leading etc. common to all runs. It may be
 ;; specialized for each run to encode the tabs that that run actually selects (via \TFBRAVO.RUN.TABSPEC

```

(CL:WHEN (IGREATERP (fetch (RUN RUNLENGTH) of RUN)
    0) ; No need for an empty piece
    (LET ((NCHARS (fetch (RUN RUNLENGTH) of RUN))
        (RUNSTART (fetch (RUN RUNSTART) of RUN))
        FATP PC)
        (SETQ PC (create PIECE
            PLEN _ NCHARS
            PLOOKS _ (\TEDIT.UNIQUIFY.CHARLOOKS (fetch (RUN RUNLOOKS) of RUN)
                TEXTOBJ)
            PPARALOOKS _ (\TEDIT.UNIQUIFY.PARALOOKS PARALOOKS TEXTOBJ)
            PPARALAST _ (fetch (RUN RUNLAST) of RUN)))
        (if (STRINGP RUNSTART)
            then ;; Run with at least one character converted to XCCS.
                (SETQ FATP (fetch (STRINGP FATSTRINGP) of RUNSTART))
                (with PIECE PC (SETQ PCONTENTS RUNSTART)
                    (SETQ PTYPE (CL:IF FATP
                        FATSTRING.PTYPE
                        THINSTRING.PTYPE))
                    (SETQ PBYTESPERCHAR (CL:IF FATP

```

```

                2
                1))
    (SETQ PBINABLE (NOT FATP))
    (SETQ PBYTELEN (UNFOLD NCHARS 2))
    (SETQ PUTF8BYTESPERCHAR 2))
  else (with PIECE PC (SETQ PCONTENTS BSTREAM)
    (SETQ PFPOS RUNSTART)
    (SETQ PTYPE THINFILE.PTYPE)
    (SETQ PBINABLE T)
    (SETQ PBYTESPERCHAR 1)
    (SETQ PBYTELEN NCHARS)
    (SETQ PUTF8BYTESPERCHAR 2)))
  (\TEDIT.INSERTPIECE PC NIL TEXTOBJ
  PC))]]

```

(\TFBRAVO.SPLIT.PARA

[LAMBDA (PARA)

; Edited 19-Feb-2025 12:15 by rmk
 ; Edited 8-Feb-2025 23:12 by rmk
 ; Edited 9-Sep-2023 21:35 by rmk
 ; Edited 22-Aug-2023 23:45 by rmk

;; The Bravo paragraph PARA may contain internal CRs or FORMS that should be broken out into separate Tedit paragraphs. All of them share
 ;; the same basic PARALOOKS, except that paragraphs after the first should have 0 for paragraph leading and first-paragraph margins. The
 ;; charlooks for each run are carried over to the splits.

;; However, we leave alone a paragraph with a special location, since we don't know how to arrange the positions of the later sub-paragraphs.

;; After the paragraphs have been split, we make another pass to see if we can decode tab t0. We could also try to merge runs with the same
 ;; charlooks, but TEDIT.PUT does that merging when the converted text is saved.

;; This smashes PARA's runs.

```

(LET ((PARALOOKS (fetch (PARA PARAFMTSPEC) of PARA))
      NEWPARAS)
  ;; RUNSTART is STRINGP for a math/hippo or other character that has been translated to XCCS
  (SETQ NEWPARAS (if [AND (fetch (PARA FORMATPTRS) of PARA)
    (FMEMB (GETPLOOKS PARALOOKS FMTSPECIALX)
      '(0 NIL))
    (FMEMB (GETPLOOKS PARALOOKS FMTSPECIALY)
      '(0 NIL))
  then [for PTR POS RUN FIRSTRUN NEWRUNLENGTH (RUNS _ (fetch (PARA RUNS) of PARA))
    in (fetch (PARA FORMATPTRS) of PARA) eachtime (SETQ POS (CDR PTR))
      (SETQ FIRSTRUN RUNS)
    when [SETQ RUN (find R in old RUNS unless (STRINGP (fetch (RUN RUNSTART)
      of R))
      suchthat (AND (IGEQ POS (fetch (RUN RUNSTART) of R))
        (ILESSP POS (IPLUS (fetch (RUN RUNSTART) of R)
          (fetch (RUN RUNLENGTH) of R))
        collect (SETQ RUNS (PROG1 (CDR RUNS)
          (RPLACD RUNS)))
          ; Chop off but keep the next batch.
        (replace (RUN RUNLAST) of RUN with T)
          ; RUN ends its paragraph
        [SETQ NEWRUNLENGTH (ADD1 (IDIFFERENCE POS (fetch (RUN RUNSTART)
          of RUN)
        (CL:UNLESS (EQ NEWRUNLENGTH (fetch (RUN RUNLENGTH) of RUN))
          ; POS pointed to the middle of RUN
        ;; Shorten RUN to characters up to and including the POS character. Subsequent characters go into
        ;; a new suffix-run that will start the next paragraph.
        (push RUNS (create RUN using RUN RUNSTART _ (ADD1 POS)
          RUNLENGTH _ (IDIFFERENCE
            (fetch (RUN RUNLENGTH) of RUN)
            NEWRUNLENGTH))
        (replace (RUN RUNLENGTH) of RUN with NEWRUNLENGTH))
        ;; Fill in RUNS here, PARALOOKS below. No more FORMATPTRS
        (create PARA
          RUNS _ FIRSTRUN)
        finally (CL:WHEN RUNS _ ; Pick up anything that's left.
          (NCONC1 $$VAL (create PARA using PARA RUNS _ RUNS)))
        ;; The first paragraph has LEADAFTER=0, all the others have 1STLEFTMAR=LEFTMAR and
        ;; LEADAFTER=LEADBEFORE=0, except that the last one keeps the original LEADAFTER. Tabs are
        ;; retained across all the runs.
        (replace (PARA PARAFMTSPEC) of (CAR $$VAL)
          with (create PARALOOKS using PARALOOKS LEADAFTER _ 0))
        (for PTAIL (NEWPARALOOKS _ (create PARALOOKS
          using PARALOOKS 1STLEFTMAR _
            (GETPLOOKS PARALOOKS LEFTMAR)
            LEADBEFORE _ 0 LEADAFTER _ 0))
        on (CDR $$VAL)
        do (replace (PARA PARAFMTSPEC) of (CAR PTAIL)
          with (CL:IF (CDR PTAIL)
            NEWPARALOOKS

```

```

                (create PARALOOKS using NEWPARALOOKS LEADAFTER _
                (GETPLOOKS PARALOOKS LEADAFTER)
                ))]
      else (CONS PARA)))
;; If t0 is the first tab specified for a run, tx is the last tab of the previous run, and t(x+1) is defined, then change t0 to t(x+1).
[for NP in NEWPARAS do (for RUN LASTTAB RUNTABS TAB0TAIL (TABDEFS _ (fetch (PARA PARATABDEFS) of NP))
  in (fetch (PARA RUNS) of NP)
  do (SETQ RUNTABS (fetch (RUN RUNTABS) of RUN))
    (if (AND (SETQ TAB0TAIL (MEMB 0 RUNTABS))
            (SETQ LASTTAB (OR (CAR (NLEFT RUNTABS 1 TAB0TAIL))
                              LASTTAB))
            (ASSOC (ADD1 LASTTAB)
                   TABDEFS))
        then (SETQ RUNTABS (COPY RUNTABS))
              (RPLACA (MEMB 0 RUNTABS)
                      (ADD1 LASTTAB))
              (SETQ LASTTAB (CAR (LAST RUNTABS)))
              (change (fetch (RUN RUNLOOKS) of RUN)
                      (create CHARLOOKS using DATUM))
              (replace (RUN RUNTABS) of RUN with RUNTABS)
        else (SETQ LASTTAB (CAR (LAST RUNTABS)]
NEWPARAS])

```

(\TFBRAVO.RUN.TABSPEC

[LAMBDA (RUN PARALOOKS)

```

; Edited 19-Feb-2025 12:16 by rmk
; Edited 8-Feb-2025 23:15 by rmk
; Edited 27-Aug-2024 22:02 by rmk
; Edited 28-Jul-2024 21:30 by rmk
; Edited 15-Mar-2024 19:42 by rmk
; Edited 22-Aug-2023 16:54 by rmk
; Edited 19-Aug-2023 15:47 by rmk

```

```

;; The CLUSERINFO contains a list of named tabs specified for this and presumably defined in the paragraph-wide PARALOOKS. This returns a
;; PARALOOKS for this run that only includes the named tabs that this run calls for.

```

```

;;
;; For the first run, the PARALOOKS is the unspecialized run for the paragraph, with empty TABSPEC. Each subsequent run is given the
;; PARALOOKS for the last run, so the tabs that were selected there are known. This is because t0 is loosely specified as picking the next tab in
;; the FMTUSERINFO after the last tab that was used in the previous run (I think). (Or perhaps as setting the next tabs TABX as the interval?)

```

```

;; Bravo's named tabs are defined for a paragraph that might consist of several runs, and each run can pick out by name just the tabs that it
;; wants. For Tedit the tabs are associated with the pieces that make up a paragraph, so we have to make sure that Bravo runs are mapped to
;; separate TEDIT pieces, and arrange it so that the pieces that correspond to separate runs have different different PARALOOKS.

```

```

;;
;; NOTE: the names in the tab definitions have been bumped up by 1 to match the names in the tab looks (e.g. (0,xxx) is (1,xxx) to correspond to
;; t1. t0 doesn't match.

```

```

(DECLARE (USEDFREE USER.CM.PARALOOKS))
(LET ([LASTTAB (CAR (LAST (GETPLOOKS PARALOOKS FMTTABS)
  (TABDEFS (FGETPLOOKS PARALOOKS FMTUSERINFO))
  (TABDEFAULT (OR (FGETPLOOKS PARALOOKS FMTDEFAULTTAB)
                  (FGETPLOOKS USER.CM.PARALOOKS FMTDEFAULTTAB)))
  (RUNTABS (fetch (RUN RUNTABS) of RUN))
  TAB TABS)
  (CL:WHEN (AND TABDEFS (NULL RUNTABS))
    (SETQ RUNTABS (CONS (CAAR TABDEFS))))
  (CL:WHEN (AND TABDEFS RUNTABS)
    (CL:WHEN (EQUAL RUNTABS '(0))
              (SETQ RUNTABS '(1 2)))
    (SETQ TABS (for TABNAME in RUNTABS
                 collect
                 ;; For t0 we try to find the tab after the one last used in the previous run.
                 (if (CDR (ASSOC TABNAME TABDEFS))
                     elseif [AND (EQ TABNAME 0)
                                   (for TDTAIL TD on TABDEFS eachtime (SETQ TD (CAR TDTAIL))
                                   when (EQ LASTTAB (CDR TD))
                                   do [SETQ TABDEFAULT (fetch TABX of (CDR (CADR TDTAIL))
                                   (RETURN (CDR (CADR TDTAIL))
                                   else (GO $$ITERATE])
                     ]
                 ]
    (SETQ TABS (SORT (for TAB in TABDEFS collect (CDR TAB))
                    (FUNCTION (LAMBDA (T1 T2)
                              (ILEQ (fetch (TAB TABX) of T1)
                                     (fetch (TAB TABX) of T2))
                    (SETQ PARALOOKS (create PARALOOKS using PARALOOKS FMTDEFAULTTAB _ TABDEFAULT FMTTABS _ TABS)))
    PARALOOKS])

```

```

;; This asserts that the tabdefs are constant across a paragraph, that the right number of tabs are on each line in a paragraph. That
;; assumption is mostly reasonable, given the paragraph splitting. The code above allows each run (piece) to have its own tab settings.
;; Although \TEDIT.FORMATLINE.UPDATELOOKS can easily be modified to allow the pieces on a line to change their tab definitions,
;; the paragraph-looks menu assumes that tabs are constant across a paragraph. So things would go bonkers.

```

(\TFBRAVO.INSTALL.PAGEFORMAT

[LAMBDA (TEXTOBJ)

; Edited 22-Sep-2023 20:04 by rmk
; Edited 17-Aug-2023 14:31 by rmk
; Edited 13-Jun-90 01:00 by mitani

:: using the information from the profile paragraphs, this function installs the pageframes

(LET (FIRSTPAGENO PAGENUMBERS PAGENUMBERX PAGENUMBERY TOPMARGIN BOTTOMMARGIN NUMBEROFCOLUMNS BETWEENCOLUMNS
COLUMNWIDTH ODDHEADINGDESC HEADINGDESC EVENHEADINGDESC HEADING.NOTONFIRSTPAGE
PAGENUMBER.NOTONFIRSTPAGE EVENHEADING.NOTONFIRSTPAGE ODDHEADING.NOTONFIRSTPAGE FIRST VERSO RECTO
PAGEPROPS ROMAN)

(for VAR

in '(FIRSTPAGENO PAGENUMBERS PAGENUMBERX PAGENUMBERY TOPMARGIN BOTTOMMARGIN NUMBEROFCOLUMNS
COLUMNWIDTH BETWEENCOLUMNS ODDHEADINGDESC HEADINGDESC EVENHEADINGDESC
HEADING.NOTONFIRSTPAGE PAGENUMBER.NOTONFIRSTPAGE EVENHEADING.NOTONFIRSTPAGE
ODDHEADING.NOTONFIRSTPAGE ROMAN)

do (SET VAR (GETTEXTPROP TEXTOBJ VAR)))

:: this assumes that TEdit does not build a default page spec. If it ever does, then this logic must change.

:: the default page frame is always built. It is sometimes built as the only page frame when there is no headings specified. However, if
:: heading is specified with the not-on-first-page specified, then we must build the default page frame simply for that reason

(CL:WHEN NUMBEROFCOLUMNS

:: if this is to be printed multicolumn then determine the column width from the numberofcolumns and the space between them. 72 is
:: points per inch, US Letter width.

(SETQ COLUMNWIDTH (IQUOTIENT (IDIFFERENCE [IDIFFERENCE (CONSTANT (TIMES 8.5 72))
(ITIMES 2 (GETTEXTPROP TEXTOBJ 'EDGEMARGIN])
(ITIMES (SUB1 NUMBEROFCOLUMNS)
BETWEENCOLUMNS))
NUMBEROFCOLUMNS)))

[SETQ PAGEPROPS `(STARTINGPAGE# ,(OR FIRSTPAGENO 1]

(PUSH PAGEPROPS 'FOLIOINFO (LIST (SELECTQ ROMAN
(x 'LOWERROMAN)
(X 'UPPERROMAN)
'ARABIC)
"" ""))

(SETQ PAGENUMBERS (NEQ PAGENUMBERS 'NO))

(SETQ FIRST (TEDIT.SINGLE.PAGEFORMAT (AND PAGENUMBERS (NOT PAGENUMBER.NOTONFIRSTPAGE))
PAGENUMBERX PAGENUMBERY NIL NIL 0 0 TOPMARGIN BOTTOMMARGIN NUMBEROFCOLUMNS
COLUMNWIDTH BETWEENCOLUMNS [COND

(HEADINGDESC (CL:UNLESS HEADING.NOTONFIRSTPAGE
(LIST HEADINGDESC)))

(ODDHEADINGDESC (CL:UNLESS ODDHEADING.NOTONFIRSTPAGE
(LIST HEADINGDESC)))

(EVENHEADINGDESC (CL:UNLESS EVENHEADING.NOTONFIRSTPAGE
(LIST EVENHEADINGDESC]

'POINTS PAGEPROPS))

(CL:WHEN (OR ODDHEADINGDESC HEADING.NOTONFIRSTPAGE PAGENUMBER.NOTONFIRSTPAGE)

(SETQ VERSO (TEDIT.SINGLE.PAGEFORMAT PAGENUMBERS PAGENUMBERX PAGENUMBERY NIL NIL 0 0 TOPMARGIN
BOTTOMMARGIN NUMBEROFCOLUMNS COLUMNWIDTH BETWEENCOLUMNS (COND

(ODDHEADINGDESC
(LIST ODDHEADINGDESC
))

(HEADINGDESC
(LIST HEADINGDESC)))

'POINTS PAGEPROPS)))

(CL:WHEN (OR EVENHEADINGDESC HEADING.NOTONFIRSTPAGE PAGENUMBER.NOTONFIRSTPAGE)

(SETQ RECTO (TEDIT.SINGLE.PAGEFORMAT PAGENUMBERS PAGENUMBERX PAGENUMBERY NIL NIL 0 0 TOPMARGIN
BOTTOMMARGIN NUMBEROFCOLUMNS COLUMNWIDTH BETWEENCOLUMNS (COND

(EVENHEADINGDESC
(LIST

EVENHEADINGDESC
))

(HEADINGDESC
(LIST HEADINGDESC)))

'POINTS PAGEPROPS)))

(TEDIT.PAGEFORMAT TEXTOBJ (TEDIT.COMPOUND.PAGEFORMAT FIRST VERSO RECTO))

)

(DEFINEQ

(\TFBRAVO.ASSERT

[LAMBDA (X Y)

; Edited 21-Oct-2024 00:27 by rmk
; Edited 9-Aug-2023 10:32 by rmk
(* gbn "19-Sep-84 21:39")

(CL:UNLESS (EQ X Y)

(\TEDIT.THELP "While parsing profile paragraph, " (CONCAT X " was expected, but " Y " was found.))))))

(\TEST.CHARACTER.LOOKS

[LAMBDA (BSTREAM)

; Edited 17-Aug-2023 09:18 by rmk
; Edited 2-Aug-2023 07:48 by rmk
; Edited 29-Jul-2023 20:25 by rmk
(* gbn "6-Feb-84 19:11")

:: returns nil until done when it returns BADLOOKS or T

(PROG (PROPERTY VALFLAG TEM (VALUE 0)

```

CHAR)
LP (while [FIXP (SETQ CHAR (FCHARACTER (BIN BSTREAM] do (SETQ VALUE CHAR)
      (SETQ VALFLAG T))
(COND
  (PROPERTY (CL:UNLESS VALFLAG
              (RETURN 'BADLOOKS))
            (SETQ PROPERTY NIL))
  (VALFLAG [SETFILEPTR BSTREAM (IDIFFERENCE (GETFILEPTR BSTREAM)
      (COND
        ([EQ CHAR (CONSTANT (CHARACTER (CHARCODE CR)
          2)
          (T 1]
          (RETURN NIL)))
      (COND
        ((SETQ TEM (SELECTQ CHAR
          ((s u b i g v S U B I G V)
          T)
          NIL))
        T
        (SETQ PROPERTY T))
        ((SETQ TEM (SELECTQ CHAR
          ((t f o)
          T)
          NIL))
        T)
        ([EQ CHAR (CONSTANT (CHARACTER (CHARCODE EOL]
          (RETURN T))
        ((NEQ CHAR ' % )
          (RETURN 'BADLOOKS]
        (SETQ VALUE 0)
        (SETQ VALFLAG NIL)
        (GO LP]))

```

(\TEST.PARAGRAPH.LOOKS

[LAMBDA (BSTREAM)

; Edited 5-Aug-2023 10:58 by rmk
; Edited 2-Aug-2023 07:46 by rmk
(* gbn "6-Feb-84 18:30")

:: test if the sequence form valid paragraph looks, do not allow empty paragraph looks

```

(PROG ((VALUE 0)
  CHAR PROPERTY TABS NONEMPTY)
LP (while [FIXP (SETQ CHAR (FCHARACTER (BIN BSTREAM] do (SETQ VALUE CHAR))
[COND
  ((SELECTQ PROPERTY
    ((l d z x e y k j c q)
    (SETQ NONEMPTY T))
    NIL) ; keep going, these are all ok
  (T (SELECTQ PROPERTY
    (% (SELECTQ CHAR
      (%) (SETQ NONEMPTY T))
      (%, (COND
        ((IGREATERP VALUE 14) ; not a legal tab no
        (RETURN NIL))
        (T (SETQ NONEMPTY T)))
      T)
    (RETURN NIL)))
    (%, (SETQ NONEMPTY T))
    ((%) (SETQ NONEMPTY T))
    (RETURN NIL) ; not a legal paragraph look
(COND
  ((AND [NEQ CHAR (CONSTANT (CHARACTER (CHARCODE EOL]
    (NEQ CHAR '\))
    (SETQ PROPERTY CHAR)
    (SETQ VALUE 0)
    (GO LP)))
  (RETURN (AND NONEMPTY CHAR])

```

)

(RPAQ? TEDIT-DEFAULT-USER.CM "TEDIT-DEFAULT-USER.CM")

(RPAQ? USER.CM.RDTBL (COPYREADTABLE))

(RPAQ? PROFILE.PARA.RDTBL (COPYREADTABLE))

(SETBRK (CHARCODE (%, %: = CR))
 NIL USER.CM.RDTBL)

(SETSEPR ' (%)
 NIL USER.CM.RDTBL)

(SETSYNTAX (CHARCODE %:)
 'SEPRCHAR PROFILE.PARA.RDTBL)

(SETSYNTAX (CHARCODE CR)

'BREAKCHAR PROFILE.PARA.RDTBL)

(SETSYNTAX (CHARCODE ^Z)

'BREAKCHAR PROFILE.PARA.RDTBL)

:: Named tabs. To be removed

(DEFINEQ

(\TFBRAVO.ADD.NAMEDTAB

[LAMBDA (RUN PARALOOKS TEXTOBJ)

; Edited 19-Feb-2025 12:17 by rmk
; Edited 8-Feb-2025 23:19 by rmk
; Edited 4-Aug-2024 18:05 by rmk
; Edited 28-Jul-2024 21:29 by rmk
; Edited 9-Sep-2023 21:44 by rmk
; Edited 18-Aug-2023 18:42 by rmk
; Edited 15-Aug-2023 00:26 by rmk
; Edited 13-Aug-2023 19:56 by rmk

:: The CLUSERINFO contains a list of tab-looks appearing in this run, and FMTUSERINFO contains the definition of tabs that have been declared
:: in this paragraph or inherited from previous paragraphs.

:: Bravo's named tabs are defined for a paragraph that might consist of several runs, and each run can pick out by name just the tabs that it
:: wants. For Tedit the tabs are associated with the pieces that make up a paragraph, so we have to make sure that Bravo runs are mapped to
:: separate TEDIT pieces, and arrange it so that the pieces that correspond to separate runs have different different PARALOOKS.

::

:: THIS IS NOT USED, TO BE REMOVED. RUNTABOFFSETS DOESN'T EXIST

(NOTUSED)

(LET ((RUNLOOKS (fetch (RUN RUNLOOKS) of RUN))
(TABDEFS (FGETPLOOKS PARALOOKS FMTUSERINFO))
(TABDEFAULT (FGETPLOOKS PARALOOKS FMTDEFAULTTAB))
(TABOFFSETS ' (fetch (RUN RUNTABOFFSETS) of RUN))
TAB TABNAMES TABS)
(SETQ TABNAMES (fetch (CHARLOOKS CLUSERINFO) of RUNLOOKS))
(CL:WHEN TABDEFS

[if TABNAMES

then (SETQ TABS (for TN in TABNAMES eachtime (add TN -1) when (SETQ TAB (CDR (ASSOC TN TABDEFS)))
unless (EQ TAB T) until (EQ TN -1) collect TAB))

elseif (CDR TABDEFS)

then ; If the run has no names, then assume that its first TAB aligns at the earliest defined tab, next aligns at the second, etc.
; Sort tabs by increasing TABX, not names.

[SETQ TABS (SORT (for TD in TABDEFS collect (CDR TD))
(FUNCTION (LAMBDA (T1 T2)
(ILEQ (fetch (TAB TABX) of T1)
(fetch (TAB TABX) of T2))

elseif (EQ 0 (CAR (CAR TABDEFS)))

then ; No name and 0, make it be the default. How else would we decide where the second tab goes?

(SETQ TABDEFAULT (fetch (TAB TABX) of (CDR TABDEFS))
(CL:WHEN (OR TABS (NEQ TABDEFAULT (FGETPLOOKS PARALOOKS FMTDEFAULTTAB)))
(SETQ PARALOOKS (create PARALOOKS using PARALOOKS FMTDEFAULTTAB _ TABDEFAULT FMTTABS _ TABS))
(\TEDIT.UNIQUIFY.PARALOOKS PARALOOKS TEXTOBJ)))
PARALOOKS])

(\TFBRAVO.COPY.NAMEDTAB

[LAMBDA (OBJ PIECE OLDCH NEWCH)

(* jds " 8-Feb-84 19:58")
(* just creates a named tab stop with the same value as the
original)

(* Note that the USING phrase will create a new TEDITOBJ as well as a TEDITUSEROBJ)

(COPY OBJ))

(\TFBRAVO.PUT.NAMEDTAB

[LAMBDA (OBJ CHARSTREAM FMTSTREAM)

(* jds " 8-Feb-84 19:59")

(* just writes the position of the tab so that a new one can be created on read)

(PRINT (IMAGEOBJPROP OBJ 'OBJECTDATUM)
CHARSTREAM])

(\TFBRAVO.GET.NAMEDTAB

[LAMBDA (CHARSTREAM TEXTSTREAM)

(* jds " 8-Feb-84 19:59")
(* should read the position, create an obj and return it)

(IMAGEOBJCREATE (RATOM CHARSTREAM)
\NAMEDTAB.IMAGEFNS])

(\NAMEDTABNYET

[LAMBDA NIL
(PROMPTPRINT "Can't do that to a named tab!"])

(* gbn "30-Dec-83 17:23")

(\NAMEDTABSIZE

```
[LAMBDA (TABOBJECT IMAGESTREAM CURRENTX RIGHTMARGIN) ; Edited 6-Aug-2023 14:24 by rmk
(* gbn "19-May-84 22:52")
  (create IMAGEBOX
    XSIZE _ (IMAX 1 (IDIFFERENCE (HCSCALE (DSPSCALE NIL IMAGESTREAM)
      (IMAGEOBJPROP TABOBJECT 'OBJECTDATUM))
      CURRENTX))
    YSIZE _ 1
    YDESC _ 0
    XKERN _ 0])
```

(\NAMEDTABPREPRINT

```
[LAMBDA (TABOBJ) ; Edited 6-Aug-2023 18:56 by rmk
(CONCAT "[TAB" (IMAGEOBJPROP TABOBJ 'TABNAME)
" ]")]
```

(\TEDIT.NAMEDTAB.INIT

```
[LAMBDA NIL ; Edited 17-Mar-2024 18:26 by rmk
; Edited 6-Aug-2023 18:59 by rmk
(* jds "22-Aug-84 14:49")
  (SETQ \NAMEDTAB.IMAGEFNS (IMAGEFNSCREATE 'NIL (FUNCTION \NAMEDTABSIZE)
    (FUNCTION \TFBRAVO.PUT.NAMEDTAB)
    (FUNCTION \TFBRAVO.GET.NAMEDTAB)
    (FUNCTION \TFBRAVO.COPY.NAMEDTAB)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION \NAMEDTABPREPRINT]))
```

```
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \NAMEDTAB.IMAGEFNS)
)
(DECLARE%: DONTEVAL@LOAD DOCOPY
(AND NIL (\TEDIT.NAMEDTAB.INIT))
)
```

FUNCTION INDEX

TEDIT.BRAVOFILE?	2	\TFBRAVO.CREATE.RUNS	8	\TFBRAVO.INSTALL.PAGEFORMAT	14
TEDITFROMBRAVO	2	\TFBRAVO.FIND.LAST.TRAILER	6	\TFBRAVO.PARSE.PARA	6
\NAMEDTABNYET	16	\TFBRAVO.FONT.FROM.CHARLOOKS	9	\TFBRAVO.PARSE.PROFILE.PARA	10
\NAMEDTABPREPRINT	17	\TFBRAVO.GET.NAMEDTAB	16	\TFBRAVO.PUT.NAMEDTAB	16
\NAMEDTABSIZI	17	\TFBRAVO.GET.USER.CM	3	\TFBRAVO.READ.CHARLOOKS	8
\TEDIT.NAMEDTAB.INIT	17	\TFBRAVO.GETPARAMS	5	\TFBRAVO.READ.PARALOOKS	7
\TEST.CHARACTER.LOOKS	14	\TFBRAVO.HANDLE.HEADING	9	\TFBRAVO.READ.USER.CM	4
\TEST.PARAGRAPH.LOOKS	15	\TFBRAVO.INIT.PAGEFORMAT	5	\TFBRAVO.READNUM?	9
\TFBRAVO.ADD.NAMEDTAB	16	\TFBRAVO.INIT.PARALOOKS	5	\TFBRAVO.RUN.TABSPEC	13
\TFBRAVO.ASSERT	14	\TFBRAVO.INSERT.PARA	11	\TFBRAVO.SPLIT.PARA	12
\TFBRAVO.COPY.NAMEDTAB	16	\TFBRAVO.INSERT.RUN	11	\TFBRAVO.USER.CM.LOOKS	4

VARIABLE INDEX

PROFILE.PARA.RDTBL	15	TEDIT-DEFAULT-USER.CM ..	15	TEDIT.INPUT.FORMATS	3	USER.CM.RDTBL	15
--------------------------	----	--------------------------	----	---------------------------	---	---------------------	----

RECORD INDEX

BRAVOFONT	1	PARA	1	RUN	1
-----------------	---	------------	---	-----------	---

MACRO INDEX

\TFBRAVO.GETFONT	1	\TOPOINTS	1
------------------------	---	-----------------	---

CONSTANT INDEX

BRAVO.TRAILER.CHARS	2
---------------------------	---
