File created: 20-Mar-2024 11:07:07 **{WMEDLEY}<library>tedit>TEDIT-PCTREE.;239**

edit by: rmk

changes to: (FNS \TEDIT.INSERTPIECES)

previous date: 17-Mar-2024 12:41:57 {WMEDLEY}<library>tedit>TEDIT-PCTREE.;238

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-PCTREECOMS
       (;; Balanced tree PIECE TABLE supporting functions
        (DECLARE%: EVAL@COMPILE DONTCOPY
               ;; \BTREEMAXCOUNT = number of children in a full node = maximum value for a node's COUNT field.
               (EXPORT (CONSTANTS (\BTREEWORDSPERSLOT 4)
                                  (\BTREEMAXCOUNT 8))
                       (RECORDS BTREENODE BTSLOT)
                       (MACROS \NTHSLOT \NEXTSLOT \PREVSLOT \LASTSLOT \FIRSTSLOT \MOVESLOT \FILLSLOT \FINDSLOT)
                       (MACROS \LASTPIECEP)
                       (I.S.OPRS inslots inpieces backpieces))
               (MACROS \INSURE.VACANT.BTREESLOT)
               (ADDVARS (INSPECTDONTSORTFIELDS BTREENODE)))
        (INITRECORDS BTREENODE)
        (INITVARS (MULTIPLE-PIECE-TABLES T))
                                                                ; Experimentation
        (GLOBALVARS MULTIPLE-PIECE-TABLES)
        (FNS \TEDIT.MAKEPCTB \TEDIT.UPDATEPCNODES \TEDIT.FIRSTPIECE \TEDIT.DELETETREE \TEDIT.INSERTTREE
             \TEDIT.LASTPIECE \TEDIT.PCTOCH \TEDIT.CHTOPC \TEDIT.SET-TOTLEN \TEDIT.MAKE.VACANT.BTREESLOT
             \TEDIT.LINKNEWPIECE \TEDIT.UNLINKPIECE \TEDIT.SPLITPIECE \TEDIT.INSERTPIECE \TEDIT.INSERTPIECES
             \TEDIT.DELETEPIECES \TEDIT.ALIGNEDPIECE)
        (COMS                                                   ; Debugging
              (FNS \TEDIT.BTVALIDATE \TEDIT.BTVALIDATE.PRINT \TEDIT.CHECK-BTREE \TEDIT.CHECK-BTREE1
                   \TEDIT.BTFAIL \TEDIT.MATCHPCS)
              (INITVARS (BTVALIDATETAGS 'DONT))
              (GLOBALVARS BTVALIDATETAGS))))

;; Balanced tree PIECE TABLE supporting functions

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(RPAQQ \BTREEWORDSPERSLOT 4)

(RPAQQ \BTREEMAXCOUNT 8)

(CONSTANTS (\BTREEWORDSPERSLOT 4)
       (\BTREEMAXCOUNT 8))
)

(DECLARE%: EVAL@COMPILE

(DATATYPE BTREENODE (;; An order-4 BTREE node for representing the piece table for TEdit.
                     DOWN1 DLEN1 DOWN2 DLEN2 DOWN3 DLEN3 DOWN4 DLEN4 DOWN5 DLEN5 DOWN6 DLEN6 DOWN7 DLEN7 DOWN8
                     DLEN8 (COUNT BYTE)                         ; # of children of this node. Must not be BITS 4 because
                                                                ; \PUTBASEPTR optimizations smash the high-order bits.
                     (UPWARD XPOINTER)                          ; Parent of this node, if any.
                     TOTLEN                                     ; Total length of this tree and subtrees

                     ))

(BLOCKRECORD BTSLOT (DOWN DLEN))
)

(/DECLAREDATATYPE 'BTREENODE
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER POINTER BYTE XPOINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '38)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \NTHSLOT MACRO ((BTREENODE N)
                          (\ADDBASE BTREENODE (UNFOLD (SUB1 N)
                                                      \BTREEWORDSPERSLOT))))

(PUTPROPS \NEXTSLOT MACRO ((SLOT)
```

```
                              (\ADDBASE SLOT \BTREEWORDSPERSLOT)))

(PUTPROPS \PREVSLOT MACRO ((SLOT)
                              (\ADDBASE SLOT (IMINUS \BTREEWORDSPERSLOT))))

(PUTPROPS \LASTSLOT MACRO ((BTNODE)
                              (\ADDBASE BTNODE (UNFOLD (SUB1 (ffetch (BTREENODE COUNT) of BTNODE))
                                                       \BTREEWORDSPERSLOT))))

(PUTPROPS \FIRSTSLOT MACRO ((BTNODE)
                              BTNODE))

(PUTPROPS \MOVESLOT MACRO ((FROMSLOT TOSLOT)
                              ;; Moves the slot information from FROMSLOT to TOSLOT, and also clears FROMSLOT.
                              (\PUTBASEPTR TOSLOT 0 (ffetch (BTSLOT DOWN) of FROMSLOT))
                                                                  ; Avoid refcnt fiddling (assumes we are uninterruptable)
                              (\PUTBASEPTR FROMSLOT 0 NIL)
                              (freplace (BTSLOT DLEN) of TOSLOT with (ffetch (BTSLOT DLEN) of FROMSLOT))
                              (freplace (BTSLOT DLEN) of FROMSLOT with 0)))

(PUTPROPS \FILLSLOT MACRO ((SLOT DWN DWNL)
                           (freplace (BTSLOT DOWN) of SLOT with DWN)
                           (freplace (BTSLOT DLEN) of SLOT with DWNL)))

(PUTPROPS \FINDSLOT MACRO [(BTNODE ITEM)
                           (find S inslots BTNODE suchthat (EQ ITEM (ffetch (BTSLOT DOWN) of S])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \LASTPIECEP MACRO (OPENLAMBDA (PC TOBJ)
                              (AND (EQ PC (ffetch (TEXTOBJ LASTPIECE) of TOBJ))
                                   PC)))
)

(DECLARE%: EVAL@COMPILE

(I.S.OPR 'inslots NIL '[SUBST (GETDUMMYVAR)
                           '$$BTBODY
                           '(bind $$BTBODY _ BODY $$BTEND declare (LOCALVARS $$BTBODY $$BTEND)
                               first (SETQ I.V. (\FIRSTSLOT $$BTBODY))
                                     (SETQ $$BTEND (\LASTSLOT $$BTBODY))
                               repeatuntil (EQ I.V. $$BTEND) by (\ADDBASE I.V. \BTREEWORDSPERSLOT]
       T)

[I.S.OPR 'inpieces NIL '(first (SETQ I.V. (\DTEST (OR BODY (GO $$OUT))
                                                  'PIECE))
                         by (\DTEST (OR (NEXTPIECE I.V.)
                                        (GO $$OUT))
                                    'PIECE]

[I.S.OPR 'backpieces NIL '(first (SETQ I.V. (\DTEST (OR BODY (GO $$OUT))
                                                    'PIECE))
                           by (\DTEST (OR (PREVPIECE I.V.)
                                          (GO $$OUT))
                                      'PIECE]
)

;; END EXPORTED DEFINITIONS

(DECLARE%: EVAL@COMPILE

(PUTPROPS \INSURE.VACANT.BTREESLOT MACRO ((BTNODE TEXTOBJ)
                                          (CL:WHEN (EQ \BTREEMAXCOUNT (ffetch (BTREENODE COUNT) of BTNODE))
                                                   (\TEDIT.MAKE.VACANT.BTREESLOT BTNODE TEXTOBJ))))
)

(ADDTOVAR INSPECTDONTSORTFIELDS BTREENODE)
)

(/DECLAREDATATYPE 'BTREENODE
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER POINTER BYTE XPOINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '38)

(RPAQ? MULTIPLE-PIECE-TABLES T)

;; Experimentation

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MULTIPLE-PIECE-TABLES)
)
```

```
(DEFINEQ
```

(\TEDIT.MAKEPCTB

```
  [LAMBDA (TEXTOBJ)                                                    ; Edited  7-Dec-2023 12:41 by rmk
                                                                       ; Edited 31-Oct-2023 10:09 by rmk
                                                                       ; Edited  8-Sep-2023 16:30 by rmk
                                                                       ; Edited 26-Apr-2023 14:03 by rmk
                                                                       ; Edited  3-Oct-2022 20:40 by rmk
                                                                       ; Edited 15-Apr-93 15:48 by jds
```

;; Refreshes TEXTOBJ to an initial empty state, e.g. for \TEDIT.INSTALL.NEWPIECES

```
    (LET ((NODE (create BTREENODE
                        COUNT _ 1
                        TOTLEN _ 0
                        DLEN1 _ 0)))
         (replace (BTREENODE DOWN1) of NODE with (create PIECE
                                                         PTYPE _ THINSTRING.PTYPE
                                                         PCONTENTS _ (CONCAT "")
                                                         PBYTESPERCHAR _ 1
                                                         PLEN _ 0
                                                         PTREENODE _ NODE
                                                         PLOOKS _ (GETTOBJ TEXTOBJ DEFAULTCHARLOOKS)
                                                         PPARALOOKS _ (GETTOBJ TEXTOBJ FMTSPEC)))
         (FSETTOBJ TEXTOBJ LASTPIECE (ffetch (BTREENODE DOWN1) of NODE))
         (FSETTOBJ TEXTOBJ HINTPC NIL)
         (FSETTOBJ TEXTOBJ TEXTLEN 0)
         (FSETTOBJ TEXTOBJ PCTB (CONS NODE])
```

(\TEDIT.UPDATEPCNODES

```
  [LAMBDA (PC DELTA TEXTOBJ)                                           ; Edited 16-Mar-2024 09:57 by rmk
                                                                       ; Edited 10-Jun-2023 00:18 by rmk
                                                                       ; Edited  8-Jun-2023 23:03 by rmk
                                                                       ; Edited 21-Apr-93 16:09 by jds
```

;; The size of the text represented by PC has grown by DELTA (negative if text is being deleted).

;; For insertions, this is called by either \TEDIT.INSERTPIECE, if a new piece is being inserted, or by \TEDIT.INSERTCH.EXTEND if the insertion is ;; a string insertion physically adjacent to a previous insertion.

;; It is assumed that PC PLEN and the corresponding DLEN in its node are consistent and correct, this updates the local TOTLEN and then ;; propagates the DELTA upwards to all ancestors.

;; This deliberately does not check for the validity of the btree, since callers are responsible for some aspects of validity (like the HINTPC). Callers ;; are responsible for bracketing this with validity checks.

```
    (bind NODE UPWARD first (SETQ NODE (ffetch (PIECE PTREENODE) of PC))
                            (SETQ UPWARD (ffetch (BTREENODE UPWARD) of NODE))
                            (add (ffetch (BTREENODE TOTLEN) of NODE)
                                 DELTA)
       while UPWARD do (add (ffetch (BTSLOT DLEN) of (\FINDSLOT UPWARD NODE))
                            DELTA)
                       (add (ffetch (BTREENODE TOTLEN) of UPWARD)
                            DELTA)
                       (SETQ NODE UPWARD)
                       (SETQ UPWARD (ffetch (BTREENODE UPWARD) of NODE))
       finally (add (ffetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
                    DELTA])
```

(\TEDIT.FIRSTPIECE

```
  [LAMBDA (TEXTOBJ)                                                    ; Edited 31-Oct-2023 19:37 by rmk
                                                                       ; Edited 11-Apr-2023 12:54 by rmk
                                                                       ; Edited 24-Aug-2022 12:45 by rmk
    (for (NODE _ (CAR (GETTOBJ TEXTOBJ PCTB))) by (ffetch (BTREENODE DOWN1) of NODE) unless (type? BTREENODE NODE)
       do ;; If we don't bottom out in a piece, something else is screwed up.  But we return NIL for the last piece, which is only there to hold the
          ;; PREV pointer to the real last piece (and maybe the initial looks).
          (RETURN (CL:UNLESS (EQ NODE (FGETTOBJ TEXTOBJ LASTPIECE))
                             NODE])
```

(\TEDIT.DELETETREE

```
  [LAMBDA (OLD PCNODE TEXTOBJ)                                         ; Edited 17-Mar-2024 00:22 by rmk
                                                                       ; Edited 31-Oct-2023 10:23 by rmk
                                                                       ; Edited 26-Oct-2023 12:50 by rmk
                                                                       ; Edited 30-May-2023 08:58 by rmk
                                                                       ; Edited  5-Sep-2022 14:24 by rmk
                                                                       ; Edited 21-Mar-95 15:29 by sybalsky:mv:envos
```

;; Old can be a piece or a node, since its length is taken from the commonly correlated DLEN.

;; NOTE: On entry the lengths of the nodes may have been adjusted to anticipate the deletion, in which case the tree is in an inconsistent state ;; (BTVALIDATE will fail). But this should restore the correctness, BTVALIDATE should be OK on exit.

```
    (FSETTOBJ TEXTOBJ HINTPC NIL)
    (if (EQ 1 (ffetch (BTREENODE COUNT) of PCNODE))
        then ;; OLD was the last child, delete the whole node
             (\TEDIT.DELETETREE PCNODE (fetch (BTREENODE UPWARD) of PCNODE)
```

```
                TEXTOBJ)
    else ;; Move each of the downs above OLDSLOT forward one slot
         (UNINTERRUPTABLY
              ;; Slide everything after OLD's slot one slot to the left
              (bind TARGET OLDSLOT (LAST _ (\LASTSLOT PCNODE)) first (SETQ OLDSLOT (\FINDSLOT PCNODE OLD))
                                                                     (CL:UNLESS OLDSLOT (SHOULDNT "Piece/node not
                                                                                                  in PCNODE"))
                                                                     (CL:WHEN (EQ OLDSLOT LAST)
                                                                ; Just shrink by one
                                                                     (\FILLSLOT OLDSLOT NIL 0)
                                                                     (GO $$OUT))
                                                                     (SETQ TARGET OLDSLOT)
                 until (EQ TARGET LAST) do (\MOVESLOT (\NEXTSLOT TARGET)
                                                      TARGET)
                                           (SETQ TARGET (\NEXTSLOT TARGET))
                 finally ;; Make PCNODE consistent with this removal, \DELETEPIECES will fix things up above.
                         ;; If we recursed up the 1-entry branch above, we ended higher up, and every thing between that dangling piece and a
                         ;; node with at least 2 entries is gone.  Those nodes are still accessible from the piece, and \UPDATEPCNODES will
                         ;; climb up and adjust them needlessly.  But it will eventually get to the ones that matter.  Otherwise, \UPDATEPCNODES
                         ;; would have to worry about nodes vs pieces.
                         (add (ffetch (BTREENODE COUNT) of PCNODE)
                              -1)))])
```

(**TEDIT.INSERTTREE**

```
[LAMBDA (NEW NEXT TEXTOBJ)                                  ; Edited  7-Dec-2023 21:08 by rmk
                                                            ; Edited 25-Nov-2023 12:24 by rmk
                                                            ; Edited 31-Oct-2023 11:04 by rmk
                                                            ; Edited  9-Jun-2023 22:33 by rmk
                                                            ; Edited 29-May-2023 23:42 by rmk
                                                            ; Edited 16-Sep-2022 12:52 by rmk
                                                            ; Edited 21-Mar-95 15:29 by sybalsky:mv:envos
    ;; Inserts NEW in front of NEXT in NEXT's parent.  NEW/NEXT are pieces or nodes. The caller guarantees that the parent has at least one empty
    ;; slot..
    ;; This should be run uninterruptably, together with whatever is needed to adjust the total length.  This by itself may leave the upper lengths in an
    ;; invalid state.
    ;;
    (FSETTOBJ TEXTOBJ HINTPC NIL)
    (LET [NEXTSLOT (PARENT (CL:IF (type? PIECE NEXT)
                                  (FGETPC NEXT PTREENODE)
                                  (ffetch (BTREENODE UPWARD) of NEXT))]
         (SETQ NEXTSLOT (\FINDSLOT PARENT NEXT))
         ;; Bump the count after  finding the slot, to open up the trailing slot
         (add (ffetch (BTREENODE COUNT) of PARENT)
              1)
         ;; Move the contents of NEXTSLOT and later slots backwards
         (for (S _ (\LASTSLOT PARENT))
              PREV by PREV do (SETQ PREV (\PREVSLOT S))
                              (\MOVESLOT PREV S)
            repeatuntil (EQ PREV NEXTSLOT))
         ;; Insert NEW into the slot now vacated by NEXT, and adjust the TOTLENs
         (\FILLSLOT NEXTSLOT NEW (if (type? BTREENODE NEW)
                                     then (freplace (BTREENODE UPWARD) of NEW with PARENT)
                                          (ffetch (BTREENODE TOTLEN) of NEW)
                                   else (FSETPC NEW PTREENODE PARENT)
                                        (PLEN NEW)))
         ;; The tree now contains the insert, whether a new piece or a split of an old one.  The counts, PLEN  and DLEN, and the total TOTLEN are
         ;; consistent at this level and below, but the caller is responsible for making sure (also uninterruptably) that any length adjustments are
         ;; propagated upwards.
         NEW])
```

(**TEDIT.LASTPIECE**

```
[LAMBDA (TEXTOBJ)                                           ; Edited 31-Oct-2023 10:20 by rmk
                                                            ; Edited 12-Apr-2023 19:23 by rmk
                                                            ; Edited 21-Aug-2022 17:13 by rmk
                                                            ; Edited 16-Aug-2022 10:16 by rmk
                                                            ; Edited 14-Apr-93 16:29 by jds
    ;; Returns the LASTPIECE by running down the right side of the B-tree.  Should be the same as (fetch LASTPIECE of TEXTOBJ).  Argument can
    ;; also be a node.
    (bind [CHILD _ (CAR (LAST (GETTOBJ TEXTOBJ PCTB] while (type? BTREENODE CHILD)
       do (SETQ CHILD (ffetch (BTSLOT DOWN) of (\LASTSLOT CHILD))) finally (RETURN CHILD])
```

(**TEDIT.PCTOCH**

```
[LAMBDA (PC TEXTOBJ)                                        ; Edited 31-Oct-2023 21:05 by rmk
                                                            ; Edited 21-Oct-2023 11:54 by rmk
```

```
                                                                    ; Edited 19-Aug-2022 22:58 by rmk
                                                                    ; Edited 18-Aug-2022 13:48 by rmk
                                                                    ; Edited  8-Aug-2022 21:50 by rmk
```

;; This returns the character number in the text stream of the first character of PC. Equivalent to mapping through the next chains from the ;; beginning, but only needs to visit the BNODES above and to the left, so more logarithmic than linear.

;; This allows for the possibility that the PCTB is a list of BTREE nodes spread out to avoid lots of big-fixp allocation and deallocation in ;; \UPDATEPCNODES.

;; The initial PC is guaranteed to have a PTREENODE--it doesn't make sense for this to be called on a piece that is not yet in a tree. Such a piece ;; does not have a char count. So the loop is executed at least once.

;; PREV starts at a piece, becomes higher BTREENODEs.

```
(bind (PREV _ PC)
      (PCNODE _ (FGETPC PC PTREENODE))
      (CHARCOUNT _ 1) do (add CHARCOUNT (for S inslots PCNODE until (EQ PREV (ffetch (BTSLOT DOWN) of S))
                                          sum (ffetch (BTSLOT DLEN) of S)))
                         (SETQ PREV PCNODE)
                         (SETQ PCNODE (ffetch (BTREENODE UPWARD) of PCNODE))
   repeatwhile PCNODE finally (RETURN (IPLUS CHARCOUNT (for TOPNODE in (FGETTOBJ TEXTOBJ PCTB)
                                                          until (EQ TOPNODE PREV)
                                                          sum (ffetch (BTREENODE TOTLEN) of TOPNODE])
```

(**TEDIT.CHTOPC**

```
  [LAMBDA (CH# TEXTOBJ TELL-PC-START?)                              ; Edited  4-Nov-2023 17:56 by rmk
                                                                    ; Edited  1-Nov-2023 23:29 by rmk
                                                                    ; Edited 13-Apr-2023 22:22 by rmk
                                                                    ; Edited 12-Apr-2023 09:49 by rmk
                                                                    ; Edited  5-Apr-2023 15:52 by rmk
                                                                    ; Edited 11-Sep-2022 13:24 by rmk
                                                                    ; Edited 15-Apr-93 16:05 by jds
```

;; Given a character # in a text object, return a pointer to the piece containing that character, else NIL.

;; The basic algorithm is a logarithmic scan of the B-tree, skipping branches at each level until the branch with CH# is reached.

;; There are 2 acceleration cases:

;; if CH# is after the current text length, the pseudo LASTPIECE is returned to the caller wo can retrieve its looks and PREV (the piece ;; containing the last actual character.

;; If the TEXTOBJ contains a HINTPC and CH# is in the range HINTPCSTARTCH# and HINTPCSTARTCH#+PLEN-1, then HINTPC is returned. ;; Others may cache that, but we cache it here too for repeated sequential calls.

;; If TELL-PC-START? is not NIL, sets the free variable START-OF-PIECE to the ch# of the piece's start.

```
(DECLARE (USEDFREE START-OF-PIECE))
(LET (HINTPC STARTCH)
     (if (IGREATERP CH# (FGETTOBJ TEXTOBJ TEXTLEN))
         then (CL:WHEN TELL-PC-START?
                   (SETQ START-OF-PIECE (ADD1 (FGETTOBJ TEXTOBJ TEXTLEN))))
              (FGETTOBJ TEXTOBJ LASTPIECE)
       elseif (AND (SETQ HINTPC (FGETTOBJ TEXTOBJ HINTPC))
                   (IGEQ CH# (SETQ STARTCH (FGETTOBJ TEXTOBJ HINTPCSTARTCH#)))
                   (ILESSP (IDIFFERENCE CH# STARTCH)
                           (PLEN HINTPC)))
         then (CL:WHEN TELL-PC-START? (SETQ START-OF-PIECE STARTCH))
              HINTPC
       elseif (ILEQ CH# 0)
         then (CL:WHEN TELL-PC-START? (SETQ START-OF-PIECE 0))
              NIL
       else (if MULTIPLE-PIECE-TABLES
                then (LET ((ALLPRIOR 0)
                           BASE-NODE START)
                          ;; When PCTB is a list of top-level BTNODES, we find the sub-tree that contains the global CH# piece, sum the
                          ;; TOTLEN's of all prior top-level nodes, retrieve the piece from the identified subtree after adjusting to its
                          ;; LOCAL#.  START-OF-PIECE, if required, is globally correct.

                          ;; This is a performance optimization for \UPDATEPCNODES in the case of building a textstream for a large file
                          ;; (longer than MAXSMALLP characters) by successive BOUT's at the end (e.g. seeing a large Lisp source file).
                          ;; Also look at the LASTPIECE case above. Also look at \INSERTPIECE.

                          (for old BASE-NODE NEXT in (FGETTOBJ TEXTOBJ PCTB)
                             do (SETQ NEXT (IPLUS ALLPRIOR (ffetch (BTREENODE TOTLEN) of BASE-NODE)))
                                (CL:WHEN (ILEQ CH# NEXT)      ; Found it
                                    (RETURN))
                                (SETQ ALLPRIOR NEXT))
                          (bind (LOCALCH# _ (IDIFFERENCE CH# ALLPRIOR))
                                (NODE _ BASE-NODE)
                                (BASE-CH# _ 1)
                               NBASE-CH# while (type? BTREENODE NODE)
                             do [SETQ NODE (for S inslots NODE
                                              do (SETQ NBASE-CH# (IPLUS BASE-CH# (ffetch (BTSLOT DLEN)
                                                                                    of S)))
                                                 (if (IGREATERP NBASE-CH# LOCALCH#)
                                                     then (RETURN (ffetch (BTSLOT DOWN) of S))
                                                   else (SETQ BASE-CH# NBASE-CH#]
                             finally ;; Eventually NODE is a piece or NIL. We cache what we just found.
                                     (FSETTOBJ TEXTOBJ HINTPC NODE)
                                     (SETQ START (IPLUS BASE-CH# ALLPRIOR))
```

```
                                    (FSETTOBJ TEXTOBJ HINTPCSTARTCH# START)
                                    (CL:WHEN TELL-PC-START? (SETQ START-OF-PIECE START))
                                    (RETURN NODE)))
            else (bind (NODE _ (CAR (FGETTOBJ TEXTOBJ PCTB)))
                       (BASE-CH# _ 1)
                       NBASE-CH# START while (type? BTREENODE NODE)
                 do [SETQ NODE (for S inslots NODE do (SETQ NBASE-CH# (IPLUS BASE-CH# (ffetch (BTSLOT DLEN)
                                                                                       of S)))
                                                      (if (IGREATERP NBASE-CH# CH#)
                                                          then (RETURN (ffetch (BTSLOT DOWN) of S))
                                                        else (SETQ BASE-CH# NBASE-CH#]
                 finally ;; Eventually NODE is a piece or NIL
                         (FSETTOBJ TEXTOBJ HINTPC NODE)
                         (FSETTOBJ TEXTOBJ HINTPCSTARTCH# BASE-CH#)
                         (CL:WHEN TELL-PC-START? (SETQ START-OF-PIECE BASE-CH#))
                         (RETURN NODE])
```

(\TEDIT.SET-TOTLEN

```
[LAMBDA (PCNODE)                                        ; Edited 21-Oct-2023 17:22 by rmk
                                                        ; Edited 15-Aug-2022 17:15 by rmk
                                                        ; Edited  9-May-93 15:40 by jds
```

;; Fix the TOTLEN field of a node to match the sum of its childrens' lengths

```
(HELP 'NOTCALLED)
(replace (BTREENODE TOTLEN) of PCNODE with (for S inslots PCNODE sum (fetch (BTSLOT DLEN) of S])
```

(\TEDIT.MAKE.VACANT.BTREESLOT

```
[LAMBDA (BTNODE TEXTOBJ)                                ; Edited 16-Mar-2024 10:23 by rmk
                                                        ; Edited  7-Dec-2023 21:08 by rmk
                                                        ; Edited 31-Oct-2023 10:32 by rmk
                                                        ; Edited 10-Jun-2023 00:13 by rmk
                                                        ; Edited 30-May-2023 12:11 by rmk
                                                        ; Edited 16-Sep-2022 12:52 by rmk
                                                        ; Edited 21-Mar-95 15:29 by sybalsky:mv:envos
```

;; Insures that BTNODE has at least one vacant slot. TEXTOBJ is needed if we have to add a new root node.

;; The intent here is that the tree is valid whenever the code is interruptable (an interrupt should never leave the tree in a trashed state.)

```
(\TEDIT.BTVALIDATE '\TEDIT.MAKE.VACANT.BTREESLOT 'START TEXTOBJ)
(CL:WHEN (EQ \BTREEMAXCOUNT (fetch (BTREENODE COUNT) of BTNODE))
```

;; All the slots of BTNODE are full. We create PREFIXNODE to hold the lower lower half of BTNODE's slots, and install that as the left sister ;; of BTNODE in its parent (perhaps first creating empty space in the parent if it is also full). The remaining slots of BTNODE are shifted down ;; to its front. We first have to make sure that the parent has room for PREFIXNODE.

;; Note that we only have to worry about count-consistency locally, since we are not really adding or subtracting, just moving things between ;; levels.

;; PREFIXNODE will hold the first half of the entries in BTNODE, SUFFIXNODE will hold the trailing entries (moved to the beginning). In the ;; end they will be the only items in BTNODE.

```
(LET (PREFIXNODE (PARENT (ffetch (BTREENODE UPWARD) of BTNODE))
              (PREFIXTOTLEN 0)
              (HALFCOUNT (FOLDLO (ffetch (BTREENODE COUNT) of BTNODE)
                                 2)))
```

;;

```
(if PARENT
    then (\INSURE.VACANT.BTREESLOT PARENT TEXTOBJ)   ; Make sure the parent has room for the coming PREFIXNODE
         (SETQ PARENT (ffetch (BTREENODE UPWARD) of BTNODE))
                                                     ; It seems that the new root parent doesn't always
                                                     ; propagate--don't know why.
  else ;; We reached the root, add a new root node containing only BTNODE and its TOTLEN. BTNODE's now has a parent with
       ;; maxcount-1 empty slots.
       (SETQ PARENT (create BTREENODE
                            COUNT _ 1
                            TOTLEN _ (fetch (BTREENODE TOTLEN) of BTNODE)))
       (\FILLSLOT (\FIRSTSLOT PARENT)
              BTNODE
              (fetch (BTREENODE TOTLEN) of BTNODE))
       (UNINTERRUPTABLY
           (replace (BTREENODE UPWARD) of BTNODE with PARENT)
           (RPLACA (OR (FMEMB BTNODE (FGETTOBJ TEXTOBJ PCTB))
                       (HELP "BTNODE NOT FOUND"))
                   PARENT)))
```

;; Tree is still valid, but PARENT how has a needed empty slot.

;;

;; We now go uninterruptable to redistribute the slots in BTNODE.

```
(SETQ PREFIXNODE (create BTREENODE
                         COUNT _ HALFCOUNT))
(UNINTERRUPTABLY
```

;; The lower entries of BTNODE become the lower entries of a new PREFIXNODE and are deleted from BTNODE. The ;; HALFCOUNT count stops the iteration.

```
                (for PRSLOT DOWN inslots PREFIXNODE as (BTSLOT _ (\FIRSTSLOT BTNODE)) by (\NEXTSLOT BTSLOT)
                   do ;; \MOVESLOT doesn't just copy, it smashes the source-slot to avoid refcount trafficking. Does this matter?
                      (\MOVESLOT BTSLOT PRSLOT)
                      (add PREFIXTOTLEN (ffetch (BTSLOT DLEN) of PRSLOT))
                      (SETQ DOWN (ffetch (BTSLOT DOWN) of PRSLOT))
                      (if (type? PIECE DOWN)
                          then (freplace (PIECE PTREENODE) of DOWN with PREFIXNODE)
                        else (freplace (BTREENODE UPWARD) of DOWN with PREFIXNODE)))
                (freplace (BTREENODE TOTLEN) of PREFIXNODE with PREFIXTOTLEN)
                ;;
                ;; Prefix node by itself is complete and valid. Adjust BTNODE to reflect that items were removed.
                (freplace (BTREENODE COUNT) of BTNODE with HALFCOUNT)
                (add (ffetch (BTREENODE TOTLEN) of BTNODE)
                     (IMINUS PREFIXTOTLEN))
                (freplace (BTSLOT DLEN) of (\FINDSLOT PARENT BTNODE) with (ffetch (BTREENODE TOTLEN) of BTNODE))
                ;; Shift the remaining slots in BTNODE to the front
                (for SUSLOT inslots BTNODE as BTSLOT inslots (\NTHSLOT BTNODE (ADD1 HALFCOUNT))
                   do (\MOVESLOT BTSLOT SUSLOT))
                ;;
                ;; Finally, add PREFIXNODE in front of BTNODE in its PARENT.
                (\TEDIT.INSERTTREE PREFIXNODE BTNODE TEXTOBJ))
            ;;
            (\TEDIT.BTVALIDATE 'TEDIT.MAKE.VACANT.BTREESLOT 'END TEXTOBJ)))])
```

(\TEDIT.LINKNEWPIECE

```
[LAMBDA (NEW NEXT TEXTOBJ)                                  ; Edited 29-May-2023 23:16 by rmk
```

;; Set up the linear-chain links to insert the piece NEW in front of the piece NEXT in its piece-chain. This doesn't deal with the btree.

;; NEXT=NIL denotes the last piece LASTPIECE of TEXTOBJ whose NEXTPIECE is NIL and whose PREVPIECE is always the last real piece of ;; the text stream.

```
(CL:UNLESS NEXT
    (SETQ NEXT (ffetch (TEXTOBJ LASTPIECE) of TEXTOBJ)))
(LET ((NEXTPREV (PREVPIECE NEXT)))
     (freplace (PIECE NEXTPIECE) of NEW with (CL:UNLESS (\LASTPIECEP NEXT TEXTOBJ)
                                                 NEXT))     ; NIL for last piece
     (freplace (PIECE PREVPIECE) of NEW with NEXTPREV)      ; Do the new piece first, interrupts OK
     (UNINTERRUPTABLY                                       ; Smash existing pieces uninterruptably
         (CL:WHEN NEXTPREV                                  ; Not at the very beginning?
             (freplace (PIECE NEXTPIECE) of NEXTPREV with NEW))
         (freplace (PIECE PREVPIECE) of NEXT with NEW))
     NEW])
```

(\TEDIT.UNLINKPIECE

```
[LAMBDA (PREV PC TEXTOBJ)                                   ; Edited 21-Oct-2023 17:24 by rmk
                                                            ; Edited 30-May-2023 00:31 by rmk
```

;; Takes PC out of the piece chain, linking prev and next around it.

```
(HELP 'NOTCALLED?)
(CL:WHEN PREV
    (freplace (PIECE NEXTPIECE) of PREV with (NEXTPIECE PC)))
(freplace (PIECE PREVPIECE) of (OR (NEXTPIECE PC)
                                   (ffetch (TEXTOBJ LASTPIECE) of TEXTOBJ))
   with PREV])
```

(\TEDIT.SPLITPIECE

```
[LAMBDA (PC CHOFFSET TEXTOBJ)                               ; Edited 17-Mar-2024 00:11 by rmk
                                                            ; Edited 28-Dec-2023 22:17 by rmk
                                                            ; Edited  7-Dec-2023 21:07 by rmk
                                                            ; Edited 25-Nov-2023 11:50 by rmk
                                                            ; Edited 31-Oct-2023 10:42 by rmk
                                                            ; Edited 27-Jul-2023 08:38 by rmk
                                                            ; Edited 30-May-2023 00:06 by rmk
                                                            ; Edited 21-Apr-93 17:49 by jds
```

;; CHOFFSET is a character offset in PC. This modifes PC if necessary so that the character at offset CHOFFST is translated to offset 0. If PC is ;; modified, a new piece is created for the characters of its truncated prefix (characters from original 0 to original (SUB1 CHNO)). The new piece is ;; linked into the piece sequence (as the PREVPIECE of PC), and the original PC (possibly shortened) is returned.

;; Whether or not a new prev piece is created, on return it is always the case that the character that was at CHOFFSET in PC is now at offset 0 in ;; PC.

```
;;
(\TEDIT.BTVALIDATE 'TEDIT.SPLITPIECE 'START TEXTOBJ)
(CL:WHEN (AND PC (IGREATERP CHOFFSET 0))                    ; Nothing to do if asking for 0.
    (FSETTOBJ TEXTOBJ HINTPC NIL)
    (\INSURE.VACANT.BTREESLOT (FGETPC PC PTREENODE)
           TEXTOBJ)                                         ; Do this before reducing PC, so tree remains valid
    (LET [(PREVPC (create PIECE using PC PPARALAST _ NIL PLEN _ CHOFFSET PBYTELEN _ (ITIMES (PBYTESPERCHAR
                                                                                            PC)
```

```
                                                                    CHOFFSET]
                                                                    ; There can be no para break before the split, as things now
                                                                    ; work.
        ;; PREVPC is the prefix before the split point of length CHOFFSET, PC will be the suffix, a shortened version of a piece that was
        ;; already in the piece tree.
        (CL:UNLESS (MEMB (PTYPE PC)
                         (CONSTANT (APPEND STRING.PTYPES FILE.PTYPES)))
                                                                    ; Dont' want the error under the UNINTERRABPTABLY.
                                                                    ; Remove when everything is good.
            (SHOULDNT "ATTEMPT TO SPLIT A NONSTRING NONFILE PIECE"))
        ;;
        (UNINTERRUPTABLY
            (SELECTC (PTYPE PC)
                (STRING.PTYPES                                      ; Adjust the offsets and lengths for strings
                    (FSETPC PREVPC PCONTENTS (SUBSTRING (PCONTENTS PC)
                                                        1 CHOFFSET))
                    (FSETPC PC PCONTENTS (SUBSTRING (PCONTENTS PC)
                                                    (ADD1 CHOFFSET))))
                (FILE.PTYPES (ADD (PFPOS PC)
                                  (ITIMES CHOFFSET (PBYTESPERCHAR PC))))
                NIL)
            ;;  PREVPC is now complete, and PC's character pointers are correct .
            ;;  PC itself must now be shortened, including its DLEN in its parent.We don't have to propagate upwards here, because this is all
            ;; length-conserving..
            (change (PLEN PC)
                    (IDIFFERENCE DATUM CHOFFSET))
            (FSETPC PC PBYTELEN (ITIMES (PBYTESPERCHAR PC)
                                        (PLEN PC)))
            (freplace (BTSLOT DLEN) of (\FINDSLOT (FGETPC PC PTREENODE)
                                                  PC)
               with (PLEN PC))
            ;; Insert PREVPC into the piece tree in front of PC.
            (\TEDIT.INSERTTREE PREVPC PC TEXTOBJ)
            (\TEDIT.LINKNEWPIECE PREVPC PC))
        (\TEDIT.BTVALIDATE '\TEDIT.SPLITPIECE 'AFTER-INSERTPIECE TEXTOBJ)))
    PC])
```

(\TEDIT.INSERTPIECE

```
  [LAMBDA (NEWPC NEXTPC TEXTOBJ)                                    ; Edited 17-Mar-2024 00:11 by rmk
                                                                    ; Edited  7-Dec-2023 21:07 by rmk
                                                                    ; Edited 31-Oct-2023 23:05 by rmk
                                                                    ; Edited  9-Jun-2023 22:40 by rmk
                                                                    ; Edited  3-Jun-2023 20:23 by rmk
                                                                    ; Edited 29-May-2023 23:23 by rmk
    ;; Insert the piece NEWPC in front of the piece NEXTPC.  At the end, NEWPC appears before NEXTPC in the piece tree, and all counts and
    ;; lengths are consistent.
    ;; The last piece LASTPIECE is always a piece in the last node whose NEXTPIECE is NIL and whose PREVPIECE is always the last real piece in
    ;; the chain. But the lastpiece has its rightful place in the tree.
    ;; Caller guarantees that the chain links of NEW can be smashed.
    (\TEDIT.BTVALIDATE '\TEDIT.INSERTPIECE 'START TEXTOBJ)
    (FSETTOBJ TEXTOBJ HINTPC NIL)
    (CL:UNLESS NEXTPC
        (SETQ NEXTPC (FGETTOBJ TEXTOBJ LASTPIECE)))
    (CL:WHEN (AND MULTIPLE-PIECE-TABLES (EQ NEXTPC (FGETTOBJ TEXTOBJ LASTPIECE)))
                                                                    ; Inserting at the very end
        (LET ((PCTB (FGETTOBJ TEXTOBJ PCTB))
              LASTTREECONS)
            ;; If the TOTLEN of the currently final top-level tree would go above MAX.SMALLP, we create a new tree that contains only the empty
            ;; last piece. The last piece is also hanging on the last branch of the previous tree, but it should never be encountered.
            (SETQ LASTTREECONS (LAST PCTB))
            (CL:WHEN (IGEQ (IPLUS (PLEN NEWPC)
                                  (ffetch (BTREENODE TOTLEN) of (CAR LASTTREECONS)))
                           (SUB1 MAX.SMALLP))
                ;; Make this uninterruptable.  We know that  NEXTPC is the zero-PLEN last piece, so no need for \UPDATEPCNODES to fix the
                ;; lengths.
                (\TEDIT.DELETETREE NEXTPC (FGETPC NEXTPC PTREENODE)
                       TEXTOBJ)
                [RPLACD LASTTREECONS
                       (SETQ LASTTREECONS
                        (CONS (create BTREENODE
                                      COUNT _ 1
                                      TOTLEN _ 0
                                      DLEN1 _ 0
                                      DOWN1 _ NEXTPC]
                (FSETPC NEXTPC PTREENODE (CAR LASTTREECONS))
                (FSETTOBJ TEXTOBJ PCTB PCTB))))
    (\INSURE.VACANT.BTREESLOT (FGETPC NEXTPC PTREENODE)
           TEXTOBJ)
```

```
    (UNINTERRUPTABLY
        (\TEDIT.INSERTTREE NEWPC NEXTPC TEXTOBJ)
        (\TEDIT.LINKNEWPIECE NEWPC NEXTPC TEXTOBJ)
        (\TEDIT.UPDATEPCNODES NEWPC (PLEN NEWPC)
               TEXTOBJ))
    (\TEDIT.BTVALIDATE '\TEDIT.INSERTPIECE 'END TEXTOBJ)
    NEWPC])
```

(\TEDIT.INSERTPIECES

```
  [LAMBDA (PIECES NEXTPC TEXTOBJ)                          ; Edited 20-Mar-2024 10:55 by rmk
                                                           ; Edited 17-Mar-2024 12:41 by rmk
                                                           ; Edited 16-Mar-2024 10:23 by rmk
                                                           ; Edited  7-Dec-2023 21:08 by rmk
                                                           ; Edited 25-Nov-2023 12:03 by rmk
                                                           ; Edited  5-Sep-2023 21:36 by rmk
                                                           ; Edited 29-Aug-2023 11:09 by rmk
                                                           ; Edited  2-Jul-2023 16:35 by rmk
                                                           ; Edited  3-Jun-2023 20:53 by rmk
                                                           ; Edited 21-May-2023 21:00 by rmk
```

;; Inserts the piece-chain PIECES in front of existing NEXTPC in TEXTOBJ. This assumes that the piece-chain is already linked, that the nextpiece ;; of the final piece in the chain is initially NIL but ends up pointing to NEXTPC (or NIL if it is the last piece).

```
    (CL:WHEN PIECES
        (TEXTOBJ! TEXTOBJ)
        (FSETTOBJ TEXTOBJ HINTPC NIL)
        (FSETTOBJ TEXTOBJ \DIRTY T)
        (CL:UNLESS NEXTPC
            (SETQ NEXTPC (FGETTOBJ TEXTOBJ LASTPIECE)))
        (for PC (PREVPC _ (PREVPIECE NEXTPC))
            inpieces PIECES do
```

;; This is a variant of \INSERTPIECE specialized for filling in an empty TEXTOBJ from a piece chain.
;; Insertion always happens before NEXTPC, and the chain-links are not smashed.

;; This may not be safe against interruptions, for a TEXTOBJ that the user already has (called from
;; \INSERTSELPIECES). The pieces that are inserted into the tree have links do so far uninserted pieces.
;; Maybe the loop itself should be uninterruptable.

```
                (UNINTERRUPTABLY
                    (\INSURE.VACANT.BTREESLOT (FGETPC NEXTPC PTREENODE)
                           TEXTOBJ)
                    (\TEDIT.INSERTTREE PC NEXTPC TEXTOBJ)
                    (\TEDIT.UPDATEPCNODES PC (PLEN PC)
                           TEXTOBJ))
            finally
```

;; PC is the final piece of the chain

```
                (CL:UNLESS (EQ NEXTPC (FGETTOBJ TEXTOBJ LASTPIECE))
                       (FSETPC PC NEXTPIECE NEXTPC))
                (FSETPC NEXTPC PREVPIECE PC)
                (CL:WHEN PREVPC (FSETPC PREVPC NEXTPIECE PIECES))
                (FSETPC PIECES PREVPIECE PREVPC)))
    PIECES])
```

(\TEDIT.DELETEPIECES

```
  [LAMBDA (SELPIECES TEXTOBJ)                              ; Edited 16-Mar-2024 10:00 by rmk
                                                           ; Edited 25-Nov-2023 12:12 by rmk
                                                           ; Edited  4-Nov-2023 23:03 by rmk
                                                           ; Edited 22-Oct-2023 11:43 by rmk
                                                           ; Edited  5-Sep-2023 22:32 by rmk
                                                           ; Edited  8-Jun-2023 23:12 by rmk
                                                           ; Edited  3-Jun-2023 22:44 by rmk
                                                           ; Edited 30-May-2023 08:57 by rmk
                                                           ; Edited 20-Apr-93 19:06 by jds
```

;; As the PC is deleted from the tree on each iteration, the original previous PREV piece is linked to PC's next, and the next PREVPIECE is linked ;; to PREV so that the tree and the links are uninterruptably consistent.

;; PREV is NIL if SPFIRST=\FIRSTPIECE; in that case the tree itself manages the connection. If SPLAST is the final actual piece (its NEXTPIECE ;; is NIL), then LASTPIECE's PREVPIECE will be updated.

;; Since the pieces are not unlinked on the fly, the tree may be invalid until all the pieces are gone.

;; This may not be entirely safe against an interrupt, which only matters on the call from \INSERTSELPIECES (otherwise the data isn't yet visible). ;; Although the tree is consistent with the remaining pieces after each deletion, the fact that we keep the SELPIECE links intact means that the ;; remaining pieces point to pieces that are no longer in the tree. We could do a little more work to incrementally chain the deleted pieces together, ;; one by one, as they are deleted--in the end they would all be out of the tree, and the deletion chain would have been reconnected. Alternatively, ;; we can make the whole loop be uninterruptable.

```
    (\TEDIT.BTVALIDATE '\TEDIT.DELETEPIECES 'BEFORE TEXTOBJ)
    (for PC PREV NEXT first (FSETTOBJ TEXTOBJ HINTPC NIL)
                           (SETQ PREV (PREVPIECE (fetch (SELPIECES SPFIRST) of SELPIECES)))
                                                           ; For incremental chain-update
                           (SETQ NEXT (OR (NEXTPIECE (fetch (SELPIECES SPLAST) of SELPIECES))
                                          (FGETTOBJ TEXTOBJ LASTPIECE)))
                           inselpieces SELPIECES
       do (UNINTERRUPTABLY
              (\TEDIT.UPDATEPCNODES PC (IMINUS (PLEN PC))
                     TEXTOBJ)
              (\TEDIT.DELETETREE PC (FGETPC PC PTREENODE)
                     TEXTOBJ)
```

```
                ;; This piece and its lengths are out of the tree, but its chain-links are still there. To keep the tree valid at each point, we incrementally
                ;; splice it out.
                (CL:WHEN PREV                                          ; Not at the very beginning
                    (FSETPC PREV NEXTPIECE (NEXTPIECE PC)))
                (FSETPC NEXT PREVPIECE PREV))
        finally ;; TEXTOBJ has forgotten the SELPIECES, now make the SELPIECES also forget they were there.
                (FSETPC (fetch (SELPIECES SPFIRST) of SELPIECES)
                        PREVPIECE NIL)
                (FSETPC (fetch (SELPIECES SPLAST) of SELPIECES)
                        NEXTPIECE NIL))
    (\TEDIT.BTVALIDATE '\TEDIT.DELETEPIECES 'AFTER TEXTOBJ])
```

(\TEDIT.ALIGNEDPIECE

```
  [LAMBDA (CHNO TEXTOBJ)                                               ; Edited 17-Mar-2024 00:27 by rmk
                                                                       ; Edited 31-Oct-2023 19:37 by rmk
                                                                       ; Edited 29-May-2023 23:48 by rmk
                                                                       ; Edited 20-May-2023 13:53 by rmk
                                                                       ; Edited  3-May-2023 18:47 by rmk
                                                                       ; Edited 21-Apr-93 17:49 by jds
    ;; CHNO is a character offset in the text.  If CHNO is not the beginning of a piece, this modifies the piecetable so that it is.  If  the piece table is
    ;; modified, a new piece is created for the characters  before CHNO  (characters from original 0 to original (SUB1 CHNO)), and the original piece is
    ;; shortened so that it no longer includes those characters.   The new piece is linked into the piece sequence.

    ;; The return is the (possibly shortened) original piece with character CHNO now at offset 0. Its PREVPIECE may or may not be new.

    (if (IGREATERP CHNO (FGETTOBJ TEXTOBJ TEXTLEN))
        then ;; Doesn't return NIL in this case, returns the last piece.
             (FGETTOBJ TEXTOBJ LASTPIECE)
      elseif (ILEQ CHNO 1)
        then (\TEDIT.FIRSTPIECE TEXTOBJ)
        else (LET (PC START-OF-PIECE)
                  (DECLARE (SPECVARS START-OF-PIECE))
                  (SETQ PC (\TEDIT.CHTOPC CHNO TEXTOBJ T))
                  (CL:UNLESS (IEQP CHNO START-OF-PIECE)                ; There can be no para break before the split, as things now
                                                                       ; work.
                      (\TEDIT.SPLITPIECE PC (IDIFFERENCE CHNO START-OF-PIECE)
                             TEXTOBJ))
                  PC])

)

;; Debugging

(DEFINEQ
```

(\TEDIT.BTVALIDATE

```
  [LAMBDA (TAG MSG TOBJ PRINT)                                         ; Edited  8-Jun-2023 22:05 by rmk
                                                                       ; Edited  3-Jun-2023 17:14 by rmk
                                                                       ; Edited 29-Aug-2022 12:10 by rmk
    (DECLARE (SPECVARS TEXTOBJ MSG TAG))
    (CL:WHEN (OR (EQMEMB TAG BTVALIDATETAGS)
                 (NULL TAG)
                 (EQMEMB 'ALL BTVALIDATETAGS))
        [LET (DEPTHHIST COUNTHIST PLENHIST (NNODES 0)
                  (NPIECES 0))
             (DECLARE (SPECVARS DEPTHHIST COUNTHIST NNODES NPIECES PLENHIST))
             (PROG1 [\TEDIT.CHECK-BTREE (if TOBJ
                                            then (TEXTOBJ TOBJ)
                                          else (OR (AND (NEQ (GETATOMVAL 'TEXTOBJ)
                                                             (EVALV 'TEXTOBJ))
                                                        (TEXTOBJ (EVALV 'TEXTOBJ)
                                                                 T))
                                                   (TEXTOBJ (WHICHW)
                                                            T)
                                                   (TEXTOBJ (EVALV 'LASTTESTSTREAM)
                                                            T)
                                                   (ERROR "NOT A TEXTOBJ"]
                    (CL:WHEN PRINT (\TEDIT.BTVALIDATE.PRINT)))])])
```

(\TEDIT.BTVALIDATE.PRINT

```
  [LAMBDA NIL                                                          ; Edited 30-May-2023 09:37 by rmk
    (DECLARE (USEDFREE DEPTHHIST COUNTHIST NNODES NPIECES PLENHIST))
    (SETQ DEPTHHIST (SORT DEPTHHIST T))
    (SETQ COUNTHIST (SORT COUNTHIST T))
    (SETQ PLENHIST (SORT PLENHIST T))
    (PRINTOUT T "Number of nodes: " NNODES T "Number of pieces: " NPIECES T "Minimum depth: " (CAAR DEPTHHIST)
           T "Maximum depth: " (CAAR (LAST DEPTHHIST))
           T "Average depth: " .F3.1 (FQUOTIENT (for DH in DEPTHHIST sum (TIMES (CAR DH)
                                                                                (CDR DH)))
                                                NPIECES)
           T "Maximum count: " (CAAR (LAST COUNTHIST))
           T "Average count: " .F1.2 (FQUOTIENT (for CH in COUNTHIST sum (TIMES (CAR CH)
```

```
                                                                    (CDR CH)))
                                          NNODES)
        T "Average PLEN: " .F5.1 (FQUOTIENT (for PLH in PLENHIST sum (TIMES (CAR PLH)
                                                                            (CDR PLH)))
                                         NPIECES)
        T "Maximum PLEN: " .I3 (CAAR (LAST PLENHIST))
        T])
```

(\TEDIT.CHECK-BTREE

```
  [LAMBDA (TEXTOBJ EMBEDDED)                                    ; Edited 17-Mar-2024 00:25 by rmk
                                                                ; Edited 21-Oct-2023 17:33 by rmk
                                                                ; Edited  7-Sep-2022 09:43 by rmk
                                                                ; Edited  4-Sep-2022 16:37 by rmk
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (for BT (LASTPIECE _ (FGETTOBJ TEXTOBJ LASTPIECE)) inside (FGETTOBJ TEXTOBJ PCTB) declare (SPECVARS LASTPIECE)
       do (\TEDIT.CHECK-BTREE1 BT 0 NIL))
    (for PC inpieces (\TEDIT.FIRSTPIECE TEXTOBJ) do (SELECTC (PTYPE PC)
                                                             (FILE.PTYPES (CL:UNLESS (STREAMP (PCONTENTS PC))
                                                                                     (\TEDIT.BTFAIL "File piece without a
                                                                                            stream" PC)))
                                                             (STRING.PTYPES
                                                                  (CL:UNLESS (STRINGP (PCONTENTS PC))
                                                                          (\TEDIT.BTFAIL "String piece without a string"
                                                                                 PC)))
                                                             (OBJECT.PTYPE (CL:UNLESS (IMAGEOBJP (PCONTENTS PC))
                                                                                      (\TEDIT.BTFAIL "Imageobject piece
                                                                                             without an object" PC)))
                                                             NIL))
    (CL:WHEN (AND (FGETTOBJ TEXTOBJ HINTPC)
                  (FGETTOBJ TEXTOBJ HINTPCSTARTCH#))
        (CL:UNLESS (IEQP (FGETTOBJ TEXTOBJ HINTPCSTARTCH#)
                         (\TEDIT.PCTOCH (FGETTOBJ TEXTOBJ HINTPC)
                                TEXTOBJ))
            (\TEDIT.BTFAIL "HINTPC is not valid" (LIST (FGETTOBJ TEXTOBJ HINTPC)
                                                      (FGETTOBJ TEXTOBJ HINTPCSTARTCH#)
                                                      (\TEDIT.PCTOCH (FGETTOBJ TEXTOBJ HINTPC)
                                                             TEXTOBJ)))))
    (CL:WHEN TEXTOBJ
        (CL:UNLESS [IEQP (FGETTOBJ TEXTOBJ TEXTLEN)
                         (for BT inside (GETTOBJ TEXTOBJ PCTB) sum (for S inslots BT
                                                                          sum (fetch (BTSLOT DLEN) of S]
            (\TEDIT.BTFAIL "TEXTLEN is inconsistent" TEXTOBJ)))
    'VALID])
```

(\TEDIT.CHECK-BTREE1

```
  [LAMBDA (NODE DEPTH PARENT)                                   ; Edited 31-Oct-2023 10:35 by rmk
                                                                ; Edited 30-May-2023 00:06 by rmk
                                                                ; Edited 27-May-2023 15:00 by rmk
                                                                ; Edited  1-Sep-2022 09:49 by rmk
                                                                ; Edited 25-Aug-2022 12:53 by rmk
                                                                ; Edited 21-Aug-2022 16:46 by rmk
    ;; Returns the TOTLEN/PLEN of NODE, after verifying that all of the nodes underneath are consistent.
    (DECLARE (USEDFREE DEPTHHIST COUNTHIST PLENHIST NNODES NPIECES TEXTOBJ LASTPIECE))
    (ADD DEPTH 1)
    (if (type? PIECE NODE)
        then [if (EQ NODE LASTPIECE)
                 then (CL:WHEN (AND (PREVPIECE LASTPIECE)
                                    (NEXTPIECE (PREVPIECE LASTPIECE)))
                               (\TEDIT.BTFAIL "(NEXT (PPREV of LASTPIECE is not NULL" LASTPIECE))
             else (CL:UNLESS (IGEQ (PLEN NODE)
                                   0)
                             (\TEDIT.BTFAIL "Negative PLEN" NODE))
                  (CL:UNLESS (OR (NEXTPIECE NODE)
                                 (EQ NODE (PREVPIECE LASTPIECE)))
                             (\TEDIT.BTFAIL "PIECE with no NEXT is not PREV of LASTPIECE" NODE))
                  (CL:UNLESS (EQ PARENT (fetch (PIECE PTREENODE) of NODE))
                             (\TEDIT.BTFAIL "Piece with wrong PTREENODE" NODE))
                  (CL:WHEN (PREVPIECE NODE)
                      (CL:UNLESS (OR (EQ NODE (NEXTPIECE (PREVPIECE NODE)))
                                     (AND (NULL (NEXTPIECE (PREVPIECE NODE)))
                                          (EQ NODE LASTPIECE)))
                                 (\TEDIT.BTFAIL "PREVPIECE is not consistent" NODE)))
                  (CL:WHEN (OR (NEXTPIECE NODE)
                               LASTPIECE)
                      (CL:UNLESS (EQ NODE (PREVPIECE (OR (NEXTPIECE NODE)
                                                         LASTPIECE)))
                                 (\TEDIT.BTFAIL "NEXTPIECE is not consistent" NODE)))]
            (add NPIECES 1)
            (add [CDR (OR (SASSOC DEPTH DEPTHHIST)
                          (CAR (PUSH DEPTHHIST (CONS DEPTH 0]
                 1)
            (add [CDR (OR (ASSOC (fetch (PIECE PLEN) of NODE)
                                 PLENHIST)
                          (CAR (PUSH PLENHIST (CONS (PLEN NODE)
```

```
                                                    0]
                  1)
            (PLEN NODE)
      else (CL:UNLESS (EQ PARENT (fetch (BTREENODE UPWARD) of NODE))
                  (\TEDIT.BTFAIL "NODE with wrong UPWARD" NODE))
           (add NNODES 1)
           (add [CDR (OR (ASSOC (fetch (BTREENODE COUNT) of NODE)
                                COUNTHIST)
                         (CAR (PUSH COUNTHIST (CONS (fetch (BTREENODE COUNT) of NODE)
                                                    0]
               1)
           (for I S from (ADD1 (fetch (BTREENODE COUNT) of NODE)) to \BTREEMAXCOUNT eachtime (SETQ S
                                                                                               (\NTHSLOT NODE I))
              unless [AND (NULL (fetch (BTSLOT DOWN) of S))
                          (MEMB (fetch (BTSLOT DLEN) of S)
                                '(0 NIL]
              do (\TEDIT.BTFAIL "Upper node entries are not empty" NODE))
           (for S DLEN CHECKLEN inslots NODE sum (SETQ DLEN (fetch (BTSLOT DLEN) of S))
                                                 (CL:UNLESS (IGEQ DLEN 0)
                                                        (\TEDIT.BTFAIL "Negative DLEN" NODE))
                                                 (CL:UNLESS (IEQP DLEN 0)
                                                                ; Could be intermediate in \INSUREVACANT.BTREESLOT
                                                     (SETQ CHECKLEN (\TEDIT.CHECK-BTREE1 (fetch (BTSLOT DOWN)
                                                                                                of S)
                                                                                            DEPTH NODE))
                                                     (CL:UNLESS (IEQP DLEN CHECKLEN)
                                                         (\TEDIT.BTFAIL "Mismatching DLEN" (LIST NODE DLEN
                                                                                                CHECKLEN))))
                                               DLEN
              finally (CL:UNLESS (IEQP (fetch (BTREENODE TOTLEN) of NODE)
                                       $$VAL)
                         (\TEDIT.BTFAIL "Mismatching TOTLEN" (LIST NODE (fetch (BTREENODE TOTLEN) of NODE)
                                                                   $$VAL)))])
```

(\TEDIT.BTFAIL

```
  [LAMBDA (STRING VAL)
    (DECLARE (USEDFREE TAG MSG))                          ; Edited 28-May-2023 08:45 by rmk
    (HELP (CONCAT (OR TAG "")
                  " "
                  (OR MSG "")
                  ": " STRING)
          VAL])
```

(\TEDIT.MATCHPCS

```
  [LAMBDA (NODE)                                          ; Edited 16-Mar-2024 11:07 by rmk
                                                          ; Edited 17-Aug-2022 19:03 by rmk
                                                          ; Edited 15-Aug-2022 23:06 by rmk
                                                          ; Edited  5-May-93 17:57 by jds
    ;; Make sure that any downs pointed to by this node point back to this node.  This is for validity testing.
    (for S DOWN inslots NODE do (SETQ DOWN (ffetch (BTSLOT DOWN) of S))
                                (if (type? PIECE DOWN)
                                    then (freplace (PIECE PTREENODE) of DOWN with NODE)
                                  elseif (type? BTREENODE DOWN)
                                    then (freplace (BTREENODE UPWARD) of DOWN with NODE])

)

(RPAQ? BTVALIDATETAGS 'DONT)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS BTVALIDATETAGS)
)
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

I.S.OPR INDEX

CONSTANT INDEX

RECORD INDEX