File created: 17-Mar-2024 12:06:12 **{WMEDLEY}<library>tedit>TEDIT-FIND.;102**

edit by: rmk

changes to: (FNS \TEDIT.BASICFIND \TEDIT.BASICFIND.BACKWARD \TEDIT.WCFIND.BACKWARD)

previous date: 15-Mar-2024 14:10:05 {WMEDLEY}<library>tedit>TEDIT-FIND.;98

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TEDIT-FINDCOMS (;; User entries
                       (FNS TEDIT.FIND TEDIT.FIND.BACKWARD TEDIT.SUBSTITUTE TEDIT.NEXT)
                       ;; Implementation
                       (FNS \TEDIT.WCFIND \TEDIT.BASICFIND \TEDIT.WCFIND.BACKWARD \TEDIT.BASICFIND.BACKWARD
                            \TEDIT.PARSE.SEARCHSTRING)))
```

;; User entries

```
(DEFINEQ
```

(TEDIT.FIND

```
  [LAMBDA (TEXTOBJ TARGETSTRING START END WILDCARDS?)
```
; Edited 19-Jun-2023 22:27 by rmk
; Edited 6-May-2018 17:34 by rmk:
; Edited 30-May-91 20:56 by jds

;; If WILDCARDS? is NIL then TEDIT.FIND returns just the start of a basic string-match.

;; Otherwise it returns a list of (MATCHSTART MATCHEND) which is the start and end char positions of the match,

;; RMK: FIND isn't undoable, FIND-AGAIN is armed on meta-g. No point in hiding a previous actual edit and then having to undo a find in order to ;; undo the intended previous event. Or maybe undoing FIND would put you back where you started?

```
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (CL:WHEN TARGETSTRING
        (SETQ TARGETSTRING (MKSTRING TARGETSTRING))
        (CL:UNLESS END
            (SETQ END (TEXTLEN TEXTOBJ)))
        (CL:UNLESS START
            (SETQ START (TEDIT.GETPOINT TEXTOBJ)))
```

;; * and # are implicitly quoted if not WILDCARDS? This could be handled simply by calling CONS instead of ;; \TEDIT.PARSE.SEARCHSTRING

```
        (CL:WHEN (ILEQ START END)
            (CL:IF WILDCARDS?
                (\TEDIT.WCFIND (ffetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
                       (\TEDIT.PARSE.SEARCHSTRING TARGETSTRING)
                       START END)
                (CAR (\TEDIT.BASICFIND (ffetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
                             TARGETSTRING START END)))))])
```

(TEDIT.FIND.BACKWARD

```
  [LAMBDA (TEXTOBJ TARGETSTRING START END WILDCARDS? AGAIN)
```
; Edited 12-Jul-2023 08:24 by rmk
; Edited 20-Jun-2023 12:12 by rmk
; Edited 18-Jun-2023 23:43 by rmk
; Edited 30-May-91 19:17 by jds

;; The search is confined to the characters between START and END. It runs backwards from END looking for the nearest match, and returns the ;; character positions of that match.

;; If WILDCARDS?, the value is the pair (MATCHSTART MATCHEND) for that match, since the caller doesn't know the length. But if not ;; WILDCARDS?, just the match-start, since the caller knows the match is (NCHARS TARGETSTRING) long. This is quirky, but that's the way it is ;; documented.

```
    (SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
    (CL:WHEN [AND TARGETSTRING (NEQ 0 (NCHARS (SETQ TARGETSTRING (MKSTRING TARGETSTRING]
        (SETQ START (IMAX 1 (OR START 1)))
        (SETQ END (IMIN (OR END (SUB1 (TEDIT.GETPOINT TEXTOBJ)))
                        (TEXTLEN TEXTOBJ)))
        (CL:WHEN AGAIN
```

;; Assume that we aren't interested in another match at the current position.

```
            (ADD END -1))
        (CL:WHEN (ILEQ START END)
            (CL:IF WILDCARDS?
                (\TEDIT.WCFIND.BACKWARD (ffetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
                       (DREVERSE (\TEDIT.PARSE.SEARCHSTRING TARGETSTRING))
                       START END)
                (CAR (\TEDIT.BASICFIND.BACKWARD (ffetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
                             TARGETSTRING START END)))))])
```

(TEDIT.SUBSTITUTE

```
  [LAMBDA (TEXTSTREAM PATTERN REPLACEMENT CONFIRM?)
```
; Edited 15-Mar-2024 14:09 by rmk
; Edited 9-Mar-2024 11:36 by rmk

```
                                                                    ; Edited  3-Mar-2024 12:24 by rmk
                                                                    ; Edited 29-Feb-2024 17:00 by rmk
                                                                    ; Edited 27-Feb-2024 08:20 by rmk
                                                                    ; Edited  6-Jan-2024 11:09 by rmk
                                                                    ; Edited 12-Nov-2023 12:29 by rmk
                                                                    ; Edited 22-Sep-2023 20:36 by rmk
                                                                    ; Edited 31-May-2023 00:04 by rmk
                                                                    ; Edited 24-May-2023 20:01 by rmk
                                                                    ; Edited 30-Mar-94 16:04 by jds
```

;; Replace all instances of PATTERN with REPLACEMENT. If CONFIRM? is non-NIL, ask before each replacement.

```
(CL:UNLESS (\TEDIT.READONLY TEXTSTREAM)
    (PROG ((TEXTOBJ (TEXTOBJ TEXTSTREAM))
           (NREPLACEMENTS 0)
           (YESLIST '("Y" "y" "yes" "YES" "T" "Yes"))
           SEARCHSTRING ABORTFLG ENDCHAR# STARTCHAR# RANGE CONFIRMFLG SEL EOLSEEN REPLACE-LEN ACTIONSTRING)
          (CL:UNLESS [SETQ SEARCHSTRING (OR PATTERN (TEDIT.GETINPUT TEXTOBJ "Search string:"
                                                                    (\TEDIT.GET.TARGET.STRING TEXTOBJ
                                                                          'TEDIT.LAST.SUBSTITUTE.STRING]
                                                                    ; If the search pattern is empty, bail out.
              (TEDIT.PROMPTPRINT TEXTOBJ "[Aborted]")
              (RETURN))
          (CL:UNLESS REPLACEMENT
              [SETQ REPLACEMENT (TEDIT.GETINPUT TEXTOBJ "Replace string:" (GETTEXTPROP TEXTOBJ
                                                                                       '
                                                                                  TEDIT.LAST.REPLACEMENT.STRING
                                                                                   ])
          (if (type? SELPIECES REPLACEMENT)
            elseif (OR (STRINGP REPLACEMENT)
                       (LITATOM REPLACEMENT))
              then (SETQ REPLACEMENT (\TEDIT.SELPIECES.FROM.STRING REPLACEMENT TEXTOBJ))
            elseif (LISTP REPLACEMENT)
              then (HELP "LISTP REPLACEMENT"))
```

;; Could be NIL or empty string, meaning just delete all occurrences.

```
          (SETQ REPLACE-LEN (fetch (SELPIECES SPLEN) of REPLACEMENT))
          (SETQ ACTIONSTRING (CL:IF (ZEROP REPLACE-LEN)
                                    "delet"
                                    "substitut"))
```

;; If a pattern is specd in the call, use the caller's confirm flag, otherwise ask for one.

```
          (SETQ CONFIRMFLG (CL:IF PATTERN
                                  CONFIRM?
                                  (MEMBER (TEDIT.GETINPUT TEXTOBJ (CONCAT "Ask before each " ACTIONSTRING
                                                                          "ion?")
                                                          "No")
                                          YESLIST)))
          (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (L-CASE ACTIONSTRING T)
                                             "ing..."))
          T)
          (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
          (\TEDIT.SHOWSEL SEL NIL)
          (\TEDIT.RESET.EXTEND.PENDING.DELETE SEL TEXTOBJ)   ; Turn off any blue pending delete
```

;; STARTCHAR# and ENDCHAR# bound each search. ENDCHAR# has to be reduced as STARTCHAR# increases, so the search stays
;; within the selection.

```
          (SETQ STARTCHAR# (GETSEL SEL CH#))
          [SETQ ENDCHAR# (IPLUS STARTCHAR# (SUB1 (GETSEL SEL DCH]
          [if CONFIRMFLG
              then ;; In this case the selection moves along, ending up at the last hit.
                   [bind PENDING.SEL CHOICE while (SETQ RANGE (TEDIT.FIND TEXTOBJ SEARCHSTRING STARTCHAR#
                                                                          ENDCHAR# T))
                      do                                            ; Show each substitution site and ask for permission
                         (SETQ PENDING.SEL (TEDIT.SETSEL TEXTOBJ (CAR RANGE)
                                                         (ADD1 (IDIFFERENCE (CADR RANGE)
                                                                            (CAR RANGE)))
                                                         'RIGHT T))
                         (\TEDIT.SHOWSEL PENDING.SEL T)
                         (TEDIT.NORMALIZECARET TEXTOBJ PENDING.SEL)
                         (SELECTQ (U-CASE (NTHCHAR (TEDIT.GETINPUT TEXTOBJ "OK to replace? ['q' quits]"
                                                                   "Yes")
                                                   1))
                             (Q (RETURN))
                             (Y                                     ; Do this one
                                (\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY REPLACEMENT 'COPY TEXTOBJ)
                                       TEXTOBJ PENDING.SEL)
                                (add NREPLACEMENTS 1)
                                (SETQ STARTCHAR# (GETSEL PENDING.SEL CHLIM))
                                                                    ; Next start, compensate for end
                                [add ENDCHAR# (IDIFFERENCE REPLACE-LEN (ADD1 (IDIFFERENCE (CADR RANGE)
                                                                                          (CAR RANGE])
                             (PROGN ;; Turn off rejected selection, search for next starting one charcter later. ENDCHAR# is still OK.
                                    (TEDIT.SHOWSEL TEXTOBJ NIL PENDING.SEL)
                                    (SETQ STARTCHAR# (ADD1 (CAR RANGE]
              else ;; No confirmation required.  Do the substitutions without showing intermediate work, collect all of the replacement events
```

```
                    (bind FIRSTHIT HITLEN HITDIFF (TOTALDIFF _ 0)
                          (SAVESEL _ (\TEDIT.COPYSEL SEL)) while (SETQ RANGE (TEDIT.FIND TEXTOBJ SEARCHSTRING
                                                                                STARTCHAR# ENDCHAR# T))
                       collect (CL:UNLESS FIRSTHIT                    ; For final line updating.
                                    (SETQ FIRSTHIT (CAR RANGE)))
                               [SETQ HITLEN (ADD1 (IDIFFERENCE (CADR RANGE)
                                                               (CAR RANGE]
                               (\TEDIT.UPDATE.SEL SEL (CAR RANGE)
                                      HITLEN
                                      'RIGHT)
                               (\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY REPLACEMENT 'COPY TEXTOBJ)
                                      TEXTOBJ SEL)
                               (add NREPLACEMENTS 1)
                               (SETQ STARTCHAR# (GETSEL SEL CHLIM))
                               (SETQ HITDIFF (IDIFFERENCE REPLACE-LEN HITLEN))
                               (add ENDCHAR# HITDIFF)
                               (add TOTALDIFF HITDIFF)
                               (\TEDIT.POPEVENT TEXTOBJ)
                       finally (CL:WHEN $$VAL
                                   ;; At least one replacement, update the lines that have changed.
                                   (\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION FIRSTHIT (IDIFFERENCE (GETSEL SEL CHLIM
                                                                                                        )
                                                                                                  FIRSTHIT))
                                   ;; We want the new selection to begin at the beginning of the original selection, somewhere before the first
                                   ;; hit, and end at the position that the prior ending moved to.  The text grew or shrank with each hit.
                                   (\TEDIT.SHOWSEL SEL NIL)
                                   (\TEDIT.UPDATE.SEL SEL (GETSEL SAVESEL CH#)
                                          (IPLUS (GETSEL SAVESEL DCH)
                                                 TOTALDIFF)
                                          'RIGHT)
                                   (\TEDIT.HISTORYADD TEXTOBJ (DREVERSE $$VAL)))]
          ;; Save the search & replacement strings to offer for next time:
               (\TEDIT.SHOWSEL SEL T)
               (PUTTEXTPROP TEXTOBJ 'TEDIT.LAST.SUBSTITUTE.STRING SEARCHSTRING)
               (PUTTEXTPROP TEXTOBJ 'TEDIT.LAST.REPLACEMENT.STRING (\TEDIT.SELPIECES.TO.STRING REPLACEMENT NIL
                                                                                               TEXTOBJ))
               (TEDIT.PROMPTPRINT TEXTOBJ (SELECTQ NREPLACEMENTS
                                                   (0 (CONCAT " No " ACTIONSTRING "ions made"))
                                                   (1 (CONCAT " 1 " ACTIONSTRING "ion made"))
                                                   (CONCAT " " (MKSTRING NREPLACEMENTS)
                                                          " " ACTIONSTRING "ions made"))
                      T)
               (RETURN NREPLACEMENTS)))])
```

(TEDIT.NEXT

```
  [LAMBDA (STREAM)                                                  ; Edited 15-Mar-2024 13:34 by rmk
                                                                    ; Edited 16-Feb-2024 23:48 by rmk
                                                                    ; Edited 14-Dec-2023 21:20 by rmk
                                                                    ; Edited 20-Jun-2023 00:05 by rmk
                                                                    ; Edited  3-May-2023 23:47 by rmk
                                                                    ; Edited 18-Apr-2023 23:46 by rmk
                                                                    ; Edited 30-May-91 20:57 by jds
    (LET ((TEXTOBJ (TEXTOBJ STREAM))
          TARGET SEL OPTION FIELDSEL)
         (SETQ SEL (TEXTSEL TEXTOBJ))
         (SETQ TARGET (TEDIT.FIND TEXTOBJ ">>*<<" NIL NIL T))       ; find the first >>delimited<< field
         (SETQ FIELDSEL (MBUTTON.FIND.NEXT.FIELD TEXTOBJ (GETSEL SEL CH#)))
                                                                    ; find the first menu-type insertion field, usually delimited with {}
         [SETQ OPTION (COND
                         [(AND TARGET FIELDSEL)                     ; take the first one
                          (COND
                             ((IGREATERP (CAR TARGET)
                                         (GETSEL FIELDSEL CH#))     ; use the {} selection
                              'FIELD)
                             (T 'TARGET]
                         (TARGET 'TARGET)
                         (FIELDSEL 'FIELD)
                         (T 'NEITHER]
         (SELECTQ OPTION
              (TARGET                                               ; Found another fill-in
                      (replace (TEXTOBJ BLUEPENDINGDELETE) of TEXTOBJ with T)
                                                                    ; Original comment: "never pending a deletion", but it is!
                      (\TEDIT.SHOWSEL SEL NIL)                      ; Set up SELECTION to be the found text
                      (\TEDIT.UPDATE.SEL SEL (CAR TARGET)
                             (IDIFFERENCE (ADD1 (CADR TARGET))
                                    (CAR TARGET))
                             'RIGHT)
                      (\TEDIT.SET.SEL.LOOKS SEL 'PENDINGDEL)        ; Always selected normally
                      (TEDIT.NORMALIZECARET TEXTOBJ)                ; And get it into the window
                      (\TEDIT.SHOWSEL SEL T))
              (FIELD                                                ; Update the selection for this textobj from the scratch sel
                                                                    ; returned from MBUTTON.FIND.NEXT.FIELD
```

```
                    (FSETTOBJ TEXTOBJ BLUEPENDINGDELETE T)
                    (\TEDIT.SHOWSEL SEL NIL)                    ; Set SELECTION to be the found text
                    (\TEDIT.UPDATE.SEL SEL (GETSEL FIELDSEL CH#)
                           (GETSEL FIELDSEL DCH)
                           'LEFT)
                    (\TEDIT.SET.SEL.LOOKS SEL 'PENDINGDEL)      ; And get it into the window
                    (TEDIT.NORMALIZECARET TEXTOBJ))
               (NEITHER (TEDIT.PROMPTPRINT TEXTOBJ "No more blanks to fill in." T)
                        (SETQ SEL NIL))
               (SHOULDNT "No legal value found in selectq in TEDIT.NEXT"))
          (CL:WHEN SEL                                          ; There really IS a selection made here, so set up the charlooks
                                                                ; for it properly.
              (FSETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL)))])

)

;; Implementation

(DEFINEQ

(\TEDIT.WCFIND
  [LAMBDA (TSTREAM TARGETLIST START END HITSTART ANCHORED)      ; Edited 19-Jun-2023 23:50 by rmk
    ;; Returns the (start end) pair of a match possibly with wild cards, where HITSTART is the first character of such a match
    (CL:UNLESS (IGREATERP START END)
        [LET (RESULT)
             (COND
                ((NULL TARGETLIST)                              ; Final match
                 (LIST (OR HITSTART (SUB1 START))
                       (SUB1 START)))
                [(EQ '%# (CAR TARGETLIST))                      ; Single-char wildcard, next segment is anchored
                 (OR (\TEDIT.WCFIND TSTREAM (CDR TARGETLIST)
                            (ADD1 START)
                            END
                            (OR HITSTART START)
                            T)
                     (CL:UNLESS ANCHORED                        ; Initial # didn't match, let it slide in this loop
                         (for S from (ADD1 START) to END when (SETQ RESULT (\TEDIT.WCFIND TSTREAM TARGETLIST S END
                                                                                          S T))
                            do (RETURN RESULT)))]
                ((EQ '* (CAR TARGETLIST))
                 ;; Variable width wildcard, not anchored so the match can slide along.
                 (\TEDIT.WCFIND TSTREAM (CDR TARGETLIST)
                        START END HITSTART))
                ((SETQ RESULT (\TEDIT.BASICFIND TSTREAM (CAR TARGETLIST)
                                   START END ANCHORED))         ; Matched a string segment, keep going
                 (\TEDIT.WCFIND TSTREAM (CDR TARGETLIST)
                        (ADD1 (CADR RESULT))
                        END
                        (OR HITSTART (CAR RESULT]))])

(\TEDIT.BASICFIND
  [LAMBDA (TSTREAM TARGETSTRING START END ANCHORED)              ; Edited 17-Mar-2024 12:06 by rmk
                                                                 ; Edited 20-Jun-2023 00:11 by rmk
                                                                 ; Edited 30-May-91 20:56 by jds
    ;; Search thru TEXTOBJ, starting where the caret is, for an exact match of TARGETSTRING.  Optionally, start the search at character START.
    ;; Returns a (startmatch endmatch) pair of character positions in TSTREAM
    (bind LASTANCHOR (NCHARS _ (NCHARS TARGETSTRING))
          (CHAR1 _ (NTHCHARCODE TARGETSTRING 1))
          (ANCHOR _ (SUB1 START)) first [SETQ LASTANCHOR (ADD1 (CL:IF ANCHORED
                                                                   ANCHOR
                                                                   (IDIFFERENCE END NCHARS))]
       eachtime (\TEDIT.TEXTSETFILEPTR TSTREAM ANCHOR)
            ;; Match failed, bump the start--single char wild-card # always matches
       while [SETQ ANCHOR (find A from (ADD1 ANCHOR) to LASTANCHOR suchthat (EQ CHAR1 (BIN TSTREAM]
       when [OR (EQ NCHARS 1)
                (for I from 2 to NCHARS always (EQ (NTHCHARCODE TARGETSTRING I)
                                                   (BIN TSTREAM]
       do (RETURN (LIST ANCHOR (IPLUS ANCHOR (SUB1 NCHARS])

(\TEDIT.WCFIND.BACKWARD
  [LAMBDA (TSTREAM TARGETLIST START END HITEND ANCHORED)         ; Edited 17-Mar-2024 11:59 by rmk
                                                                 ; Edited 20-Jun-2023 13:52 by rmk
    ;; Returns the (start end) pair of a match possibly with wild cards, where HITEND is the last character of such a match
    (LET (RESULT)
         (COND
            ((NULL TARGETLIST)                                   ; Final match
             (LIST (ADD1 (\TEDIT.TEXTGETFILEPTR TSTREAM))
                   (OR HITEND END)))
            [(EQ '%# (CAR TARGETLIST))                           ; Single-char wildcard, next segment is anchored
```

```
            (OR (\TEDIT.WCFIND.BACKWARD TSTREAM (CDR TARGETLIST)
                       START
                       (SUB1 END)
                       (OR HITEND END)
                       T)
                (CL:UNLESS ANCHORED                                    ; Initial # didn't match, let it slide in this loop
                    (for E from (SUB1 END) to START by -1 when (SETQ RESULT (\TEDIT.WCFIND.BACKWARD TSTREAM
                                                                              TARGETLIST START E E T))
                       do (RETURN RESULT)))]
           ((EQ '* (CAR TARGETLIST))
            ;; Variable width wildcard, not anchored so the match can slide along.
            (\TEDIT.WCFIND.BACKWARD TSTREAM (CDR TARGETLIST)
                   START END HITEND))
           ((SETQ RESULT (\TEDIT.BASICFIND.BACKWARD TSTREAM (CAR TARGETLIST)
                                START END ANCHORED))                   ; Matched a string segment, keep going
            (\TEDIT.WCFIND.BACKWARD TSTREAM (CDR TARGETLIST)
                   START
                   (SUB1 (CAR RESULT))
                   (OR HITEND (CADR RESULT])
```

(\TEDIT.BASICFIND.BACKWARD

```
[LAMBDA (TSTREAM TARGETSTRING START END ANCHORED)                      ; Edited 17-Mar-2024 12:06 by rmk
                                                                       ; Edited 12-Jul-2023 08:14 by rmk
                                                                       ; Edited 23-Apr-2023 12:42 by rmk
```

;; Returns a (Startmatch Endmatch) pair of character positions in TSTREAM that denote the nearest occurrence of TARGETSTRING whose first ;; character is at or ahead of START and whose last character is at or before END.

;; A better interface would return a selection for the string-match, but we repeat the pair interface that is documented for forward search.

;; Note that caller must decrement END in subsequent calls to avoid looping on the same match.

;;

;; The last target character first matches at END. Setting the initial ANCHOR one past END and going into the anchor backup loop won't work if ;; END points to the last character in the stream--the \TEXTSETFILEPTR would be out of bounds. So the first anchor-match has to be special, by ;; setting the fileptr at END and peeking.

```
[SETQ END (IMIN END (TEXTLEN (TEXTOBJ TSTREAM]
(bind ANCHOR LASTANCHOR (NCHARS1 _ (SUB1 (NCHARS TARGETSTRING)))
      (CHARN _ (NTHCHARCODE TARGETSTRING -1))
   first ;; NCHARS1 because the last character is matched separately.
      (CL:WHEN (ILESSP (IDIFFERENCE END START)
                       NCHARS1)                                      ; Too few characters
          (RETURN NIL))
      (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 END))
      (CL:WHEN [AND (EQ CHARN (\TEDIT.TEXTPEEKBIN TSTREAM))
                    (OR (EQ NCHARS1 0)
                        (for I from NCHARS1 to 1 by -1 always (EQ (NTHCHARCODE TARGETSTRING I)
                                                                  (\TEDIT.TEXTBACKFILEPTR TSTREAM]
          (RETURN (LIST (IDIFFERENCE END NCHARS1)
                        END)))
      (CL:WHEN ANCHORED                                              ; Anchored at END, didn't match
          (RETURN NIL))
      (SETQ ANCHOR (SUB1 END))
      (SETQ LASTANCHOR (IPLUS START NCHARS1))
   eachtime (\TEDIT.TEXTSETFILEPTR TSTREAM ANCHOR)                   ; The filepos one before the last CHARN match
      (ADD ANCHOR -1)                                                ; For next attempt
   while (find old ANCHOR from ANCHOR to LASTANCHOR by -1 suchthat (EQ CHARN (\TEDIT.TEXTBACKFILEPTR TSTREAM)))
   when [OR (EQ NCHARS1 0)
            (for I from NCHARS1 to 1 by -1 always (EQ (NTHCHARCODE TARGETSTRING I)
                                                      (\TEDIT.TEXTBACKFILEPTR TSTREAM]
   do (ADD ANCHOR 1)
      (RETURN (LIST (IDIFFERENCE ANCHOR NCHARS1)
                    ANCHOR])
```

(\TEDIT.PARSE.SEARCHSTRING

```
[LAMBDA (TARGETSTRING)                                                 ; Edited 19-Jun-2023 16:42 by rmk
                                                                       (* jds "31-Jan-84 13:26")
```

;; Quote ls an escape if it comes before a wild card. ''# would match ' in front of literal .

```
(for TTAIL C SEG on (CHCON TARGETSTRING) do (SETQ C (CAR TTAIL))
                                            (SELCHARQ C
                                                 (%' (if (MEMB (CADR TTAIL)
                                                               (CHARCODE (%# *)))
                                                         then (POP TTAIL)
                                                              (PUSH SEG (CAR TTAIL))
                                                       else (PUSH SEG C)))
                                                 (%# (CL:WHEN SEG
                                                          (push $$VAL (CONCATCODES (DREVERSE SEG))))
                                                     (push $$VAL (CHARACTER C))
                                                     (SETQ SEG NIL))
                                                 (* (CL:UNLESS (EQ (CAR $$VAL)
                                                                   '*)
                                                               ; Reduce adjacent *s to one.
                                                       (CL:WHEN SEG
```

```
                                                    (push $$VAL (CONCATCODES (DREVERSE SEG))))
                                                (CL:UNLESS $$VAL
                                                     ; Ignore leading *
                                                     (push $$VAL (CHARACTER C)))
                                                (SETQ SEG NIL)))
                                            (PUSH SEG C))
        finally [if SEG
                    then (PUSH $$VAL (CONCATCODES (DREVERSE SEG)))
                  else                                          ; Ignore trailing *
                       (SETQ $$VAL (find VTAIL on $$VAL suchthat (NEQ (CAR $$VAL)
                                                                      '*]
                (RETURN (CL:IF $$VAL
                               (DREVERSE $$VAL)
                               TARGETSTRING)])

)
```

FUNCTION INDEX