*File created:* 6-Mar-2025 20:18:04 **{WMEDLEY}<library>TEDIT>TEDIT-FIND.;138**

*edit by:* rmk

*changes to:* (FNS TEDIT.SUBSTITUTE)

*previous date:* 17-Feb-2025 12:25:36 {WMEDLEY}<library>TEDIT>TEDIT-FIND.;136

*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 (TSTREAM TARGET START END WILDCARDS?)                    ; Edited 10-May-2024 21:55 by rmk
                                                                   ; Edited 24-Apr-2024 23:47 by rmk
                                                                   ; 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 TSTREAM (TEXTSTREAM TSTREAM))
(CL:WHEN TARGET
```

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

```
    [if (IMAGEOBJP TARGET)
        then (TEDIT.FIND.OBJECT TSTREAM TARGET START END)
      elseif [NEQ 0 (NCHARS (SETQ TARGET (MKSTRING TARGET]
        then (CL:UNLESS END
                  (SETQ END (FGETTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)
                                      TEXTLEN)))
             (CL:UNLESS START
                 (SETQ START (TEDIT.GETPOINT TSTREAM)))
             (CL:WHEN (ILEQ START END)
                 (CL:IF WILDCARDS?
                     (\TEDIT.WCFIND TSTREAM (\TEDIT.PARSE.SEARCHSTRING TARGET)
                            START END)
                     (CAR (\TEDIT.BASICFIND TSTREAM TARGET START END))))])])
```

## (TEDIT.FIND.BACKWARD

```
  [LAMBDA (TSTREAM TARGET START END WILDCARDS? AGAIN)              ; Edited 19-May-2024 12:07 by rmk
                                                                   ; Edited 10-May-2024 22:00 by rmk
                                                                   ; Edited 24-Apr-2024 23:43 by rmk
                                                                   ; 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 TSTREAM (TEXTSTREAM TSTREAM))
(CL:WHEN TARGET
    [if (IMAGEOBJP TARGET)
        then (TEDIT.FIND.OBJECT.BACKWARD TSTREAM TARGET START END AGAIN)
      elseif [NEQ 0 (NCHARS (SETQ TARGET (MKSTRING TARGET]
        then (SETQ START (IMAX 1 (OR START 1)))
             (SETQ END (IMIN (OR END (SUB1 (TEDIT.GETPOINT TSTREAM)))
                             (FGETTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)
                                    TEXTLEN)))
             (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 TSTREAM (\TEDIT.PARSE.SEARCHSTRING TARGET T)
                           START END)
                    (CAR (\TEDIT.BASICFIND.BACKWARD TSTREAM TARGET START END))))])])
```

## (TEDIT.SUBSTITUTE

```
  [LAMBDA (TSTREAM PATTERN REPLACEMENT CONFIRM?)                    ; Edited  6-Mar-2025 20:17 by rmk
                                                                    ; Edited  8-Dec-2024 15:47 by rmk
                                                                    ; Edited 26-Nov-2024 23:49 by rmk
                                                                    ; Edited 15-Aug-2024 09:20 by rmk
                                                                    ; Edited 14-Jul-2024 00:24 by rmk
                                                                    ; Edited  7-Jul-2024 11:46 by rmk
                                                                    ; Edited 29-Jun-2024 10:49 by rmk
                                                                    ; Edited 18-May-2024 23:03 by rmk
                                                                    ; Edited  9-Mar-2024 11:36 by rmk
                                                                    ; Edited 12-May-2024 21:11 by rmk
                                                                    ; Edited 15-Mar-2024 14:09 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 30-Mar-94 16:04 by jds
    (SETQ TSTREAM (TEXTSTREAM TSTREAM))
    ;; Replace all instances of PATTERN with REPLACEMENT.  If CONFIRM?  is non-NIL, ask before each replacement.
    (CL:UNLESS (\TEDIT.READONLY TSTREAM)
        (RESETLST
            (PROG ((TEXTOBJ (TEXTOBJ TSTREAM))
                   (NREPLACEMENTS 0)
                   (YESLIST '("Y" "y" "yes" "YES" "T" "Yes"))
                   SEARCHSTRING ABORTFLG ENDCHAR# STARTCHAR# RANGE CONFIRMFLG SEL EOLSEEN REPLACE-LEN
                   ACTIONSTRING)
                  ;; Don't call \TEDIT.GET.TARGET.STRING because it might pick the search-domain (current selection) as the search string.  If the
                  ;; search pattern is empty, bail out.
                  (CL:UNLESS SEARCHSTRING
                      [SETQ SEARCHSTRING (OR PATTERN (TEDIT.GETINPUT TEXTOBJ "Search string:"
                                                                     (GETTEXTPROP TEXTOBJ 'TEDIT.LAST.SUBSTITUTE.STRING])
                      (CL:UNLESS [OR REPLACEMENT (SETQ REPLACEMENT (TEDIT.GETINPUT TEXTOBJ "Replace string:"
                                                                                   (GETTEXTPROP TEXTOBJ
                                                                                          'TEDIT.LAST.REPLACEMENT.STRING]
                          (TEDIT.PROMPTPRINT TEXTOBJ "[Aborted]")
                          (RETURN))
                      [RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ "Substitute")
                            '(PROGN (\TEDIT.MARKINACTIVE OLDVALUE]
                      (if (type? SELPIECES REPLACEMENT)
                        elseif (OR (STRINGP REPLACEMENT)
                                   (LITATOM REPLACEMENT))
                          then (SETQ REPLACEMENT (\TEDIT.SELPIECES.FROM.STRING REPLACEMENT TEXTOBJ))
                        else (RETURN NIL))
                  ;; Could be NIL or empty string, meaning just delete all occurrences.
                  (SETQ REPLACE-LEN (GETSPC REPLACEMENT SPLEN))
                  (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 (FGETTOBJ TEXTOBJ SEL))
                  (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
                  (\TEDIT.RESET.EXTEND.PENDING.DELETE 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# (CL:IF (ZEROP (GETSEL SEL DCH))
                                        (GETTOBJ TEXTOBJ TEXTLEN)
                                        (IPLUS STARTCHAR# (SUB1 (GETSEL SEL DCH))))]
                  ;; NOTE:  SEARCHSTRING may contain wild cards, so the hits may be of different lengths.
                  [if CONFIRMFLG
                      then ;; In this case the selection moves along, ending up at the last hit.
                           (bind (LASTSEL _ (\TEDIT.COPYSEL SEL)) while (SETQ RANGE
                                                                              (TEDIT.FIND TEXTOBJ SEARCHSTRING
                                                                                     STARTCHAR# ENDCHAR# T))
                              do                                  ; Show each substitution site and ask for permission
                                 (\TEDIT.UPDATE.SEL SEL (CAR RANGE)
```

```
                                NIL
                                'RIGHT
                                'PENDINGDEL
                                (ADD1 (CADR RANGE)))
                        (\TEDIT.FIXSEL SEL TEXTOBJ)
                        (\TEDIT.SHOWSEL SEL T TEXTOBJ)
                        (TEDIT.NORMALIZECARET TEXTOBJ SEL)
                        [SELECTQ (U-CASE (NTHCHAR (TEDIT.GETINPUT TEXTOBJ "OK to replace? ['q' quits]"
                                                                  "Yes")
                                                  1))
                             (Q (GO $$OUT))
                             (Y                                   ; Do this one
                                (\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY REPLACEMENT
                                                                                 'COPY TEXTOBJ)
                                       TEXTOBJ SEL)
                                (\TEDIT.COPYSEL SEL LASTSEL)
                                                                   ; This may be where we end up
                                (add NREPLACEMENTS 1)
                                (SETQ STARTCHAR# (GETSEL 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 SEL NIL TEXTOBJ)
                                    (SETQ STARTCHAR# (ADD1 (CAR RANGE]
                        finally (\TEDIT.COPYSEL LASTSEL SEL))
                else ;; No confirmation required.  Do the substitutions without showing intermediate work, collect all of the replacement
                     ;; events
                     (bind FIRSTHIT HITLAST HITLEN HITDIFF (TOTALDIFF _ 0)
                           EVENTS while (SETQ RANGE (TEDIT.FIND TEXTOBJ SEARCHSTRING STARTCHAR# ENDCHAR# T))
                        do (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.FIXSEL SEL TEXTOBJ)
                           (\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY REPLACEMENT 'COPY TEXTOBJ)
                                  TEXTOBJ SEL)
                           (push EVENTS (\TEDIT.POPEVENT TEXTOBJ))
                                                                   ; Collect the events for a single composite
                           (add NREPLACEMENTS 1)
                           (SETQ STARTCHAR# (GETSEL SEL CHLIM))
                           (SETQ HITLAST STARTCHAR#)
                           (SETQ HITDIFF (IDIFFERENCE REPLACE-LEN HITLEN))
                           (add ENDCHAR# HITDIFF)
                           (add TOTALDIFF HITDIFF)
                        finally (CL:UNLESS (EQ NREPLACEMENTS 0)
                                    ;; At least one replacement, update the lines that have changed.  We have to calculate how many of
                                    ;; the original characters have "changed" by adding the TOTALDIFF to the final position of the last
                                    ;; character of the last hit.  Might be better if UPDATELINES took a lastchangechar.
                                    (if (IGREATERP TOTALDIFF 0)
                                        then (\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION FIRSTHIT
                                                  (IDIFFERENCE (IPLUS (FGETSEL SEL CHLIM)
                                                                      TOTALDIFF)
                                                               FIRSTHIT))
                                      elseif (ILESSP TOTALDIFF 0)
                                        then (\TEDIT.UPDATE.LINES TEXTOBJ 'DELETION FIRSTHIT
                                                  (IDIFFERENCE (IDIFFERENCE (FGETSEL SEL CHLIM)
                                                                            TOTALDIFF)
                                                               FIRSTHIT))
                                      else (\TEDIT.UPDATE.LINES TEXTOBJ 'CHANGED FIRSTHIT
                                                (IDIFFERENCE (FGETSEL SEL CHLIM)
                                                             FIRSTHIT)))
                                    ;; Not clear what the final selection should be, if there are multiple changes.  The original selection? A
                                    ;; selection that goes from the beginning of the first subsitution to the end of the last (as here)?  Or just
                                    ;; the selection of the last substitution?
                                    (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
                                    (\TEDIT.UPDATE.SEL SEL FIRSTHIT (IDIFFERENCE HITLAST FIRSTHIT)
                                           'RIGHT)
                                    (\TEDIT.FIXSEL SEL TEXTOBJ)
                                    (\TEDIT.HISTORYADD.COMPOSITE TEXTOBJ EVENTS))]
;; Save the search & replacement strings to offer for next time:
     (\TEDIT.SHOWSEL SEL T TEXTOBJ)
     (TEDIT.NORMALIZECARET TSTREAM SEL)
     (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 (TSTREAM)                                                    ; Edited 15-Feb-2025 18:08 by rmk
                                                                       ; Edited 21-Oct-2024 00:40 by rmk
                                                                       ; Edited  7-Jul-2024 11:47 by rmk
                                                                       ; Edited 18-May-2024 16:23 by rmk
                                                                       ; Edited 12-May-2024 21:10 by rmk
                                                                       ; Edited 16-Feb-2024 23:48 by rmk
                                                                       ; Edited 15-Mar-2024 13:34 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 TSTREAM))
          TARGET SEL OPTION FIELDSEL)
         (SETQ SEL (TEXTSEL TEXTOBJ))
         (SETQ TARGET (TEDIT.FIND TEXTOBJ ">>*<<" NIL NIL T))          ; find the first >>delimited<< field
         (SETQ FIELDSEL (TEDIT.FIND TEXTOBJ "{*}" NIL NIL T))          ; 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 TEXTOBJ)                  ; Set up SELECTION to be the found text
                     (\TEDIT.UPDATE.SEL SEL (CAR TARGET)
                            (IDIFFERENCE (ADD1 (CADR TARGET))
                                   (CAR TARGET))
                            'RIGHT
                            'PENDINGDEL)
                     (\TEDIT.FIXSEL SEL TEXTOBJ)                       ; Always selected normally
                     (\TEDIT.SHOWSEL SEL T TEXTOBJ)
                     (TEDIT.NORMALIZECARET TEXTOBJ)                    ; And get it into the window

                     )
             (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 TEXTOBJ)                   ; Set SELECTION to be the found text
                    (\TEDIT.UPDATE.SEL SEL (GETSEL FIELDSEL CH#)
                           (GETSEL FIELDSEL DCH)
                           'LEFT
                           'PENDINGDEL)                                ; And get it into the window
                    (\TEDIT.FIXSEL SEL TEXTOBJ)
                    (\TEDIT.SHOWSEL SEL T TEXTOBJ)
                    (TEDIT.NORMALIZECARET TEXTOBJ))
             (NEITHER (TEDIT.PROMPTPRINT TEXTOBJ "No more blanks to fill in." T)
                      (SETQ SEL NIL))
             (\TEDIT.THELP "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)                               ; Edited 26-Jun-2024 08:04 by rmk
                                                                       ; Edited 23-Jun-2024 12:00 by rmk
                                                                       ; Edited 19-May-2024 23:46 by rmk
                                                                       ; Edited  3-May-2024 07:11 by rmk
                                                                       ; Edited 29-Apr-2024 20:45 by rmk
                                                                       ; Edited 17-Mar-2024 11:59 by rmk
                                                                       ; Edited 20-Jun-2023 13:52 by rmk
    ;; Returns the (start end) pair of the nearest match somewhere at or after START, possibly with wild cards. The basic-find does fast search of
    ;; simple strings.  This is all about backtracking to advance the search on failure, and for wild cards. Note that *'s do not appear on the edges.
    (CL:WHEN TARGETLIST
        [bind STACK CONFIG HITSTART ANCHORED RESULT TARGETTAIL TARGET (TOPSTART _ (SUB1 START))
```

```
          do (SETQ CONFIG (pop STACK))
             (if CONFIG
                 then (SETQ START (pop CONFIG))
                      (SETQ TARGETTAIL (pop CONFIG))
                      (SETQ HITSTART (pop CONFIG))
                      (SETQ ANCHORED (pop CONFIG))
               elseif (IGEQ TOPSTART END)
                 then (RETURN NIL)                                  ; No more, failed
               else (add TOPSTART 1)                                ; First time or outer advance
                    (SETQ START TOPSTART)
                    (SETQ TARGETTAIL TARGETLIST)
                    (SETQ HITSTART NIL)
                    (SETQ ANCHORED NIL))
             (SETQ TARGET (CAR TARGETTAIL))
             (SELECTQ TARGET
                 (%# (CL:UNLESS (CDR TARGETTAIL)
                          (RETURN (LIST (OR HITSTART START)
                                        START)))
                     (CL:WHEN (ILEQ START END)                      ; If we are unanchored, slipping continues
                         (push STACK (LIST (ADD1 START)
                                           (CDR TARGETTAIL)
                                           (OR HITSTART START)
                                           ANCHORED))))
                 (* ;; Unanchored config for the tail that starts here.
                    (push STACK (LIST START (CDR TARGETTAIL)
                                      HITSTART NIL)))
                 (if (SETQ RESULT (\TEDIT.BASICFIND TSTREAM TARGET START END ANCHORED))
                     then (CL:UNLESS (CDR TARGETTAIL)               ; Success!
                              (RETURN (LIST (OR HITSTART (CAR RESULT))
                                            (CADR RESULT))))
                          (SETQ START (ADD1 (CADR RESULT)))         ; Next target
                          (CL:WHEN (ILEQ START END)
                              [push STACK (LIST START (CDR TARGETTAIL)
                                                (OR HITSTART (CAR RESULT])
                   elseif (NOT ANCHORED)
                     then (RETURN NIL])])
```

## (**TEDIT.BASICFIND**

```
  [LAMBDA (TSTREAM TARGETSTRING START END ANCHORED)                 ; Edited 17-Feb-2025 12:24 by rmk
                                                                    ; Edited 23-Jun-2024 12:03 by rmk
                                                                    ; Edited 22-Jun-2024 12:01 by rmk
                                                                    ; Edited 19-May-2024 23:18 by rmk
                                                                    ; 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 TSTREAM for an exact match of TARGETSTRING.

;; Returns a (startmatch endmatch) pair of character positions in TSTREAM

```
    (bind LASTANCHOR (NCHARS _ (NCHARS TARGETSTRING))
          (ANCHOR _ (SUB1 START)) first (CL:WHEN (ZEROP NCHARS)
                                                 (RETURN NIL))
                                        [SETQ LASTANCHOR (ADD1 (CL:IF ANCHORED
                                                                      ANCHOR
                                                                      (IDIFFERENCE END NCHARS))]
                                        ;; LASTANCHOR protects us from running into the EOF

      eachtime (CL:WHEN (IGEQ ANCHOR LASTANCHOR)
                        (RETURN NIL))
               (\TEDIT.TEXTSETFILEPTR TSTREAM ANCHOR)
               (add ANCHOR 1)                                       ; Move the anchor up 1
               ;; Match failed, bump the start--single char wild-card # always matches

      when (for I from 1 do (CL:UNLESS (EQ (NTHCHARCODE TARGETSTRING I)
                                           (BIN TSTREAM))
                                    (RETURN NIL))
                            (CL:WHEN (EQ I NCHARS)                  ; Matched the last char
                                (RETURN T)))
      do (FSETTOBJ (GETTSTR TSTREAM TEXTOBJ)
                   LASTARROWX NIL)
         (RETURN (LIST ANCHOR (IPLUS ANCHOR (SUB1 NCHARS])
```

## (**TEDIT.WCFIND.BACKWARD**

```
  [LAMBDA (TSTREAM TARGETLIST START END)                            ; Edited 26-Jun-2024 08:05 by rmk
                                                                    ; Edited 23-Jun-2024 12:02 by rmk
                                                                    ; Edited 19-May-2024 23:46 by rmk
                                                                    ; Edited  3-May-2024 07:11 by rmk
                                                                    ; Edited 29-Apr-2024 20:45 by rmk
                                                                    ; Edited 17-Mar-2024 11:59 by rmk
                                                                    ; Edited 20-Jun-2023 13:52 by rmk
```

;; Returns the (start end) pair of the nearest match somewhere at or after START, possibly with wild cards. The basic-find does fast search of ;; simple strings. This is all about backtracking to advance the search on failure, and for wild cards. Note that *'s do not appear on the edges.

```
    (CL:WHEN TARGETLIST
        [bind STACK CONFIG HITEND ANCHORED RESULT TARGETTAIL TARGET (TOPEND _ (ADD1 END))
           do (SETQ CONFIG (pop STACK))
              (if CONFIG
                  then (SETQ END (pop CONFIG))
                       (SETQ TARGETTAIL (pop CONFIG))
                       (SETQ HITEND (pop CONFIG))
                       (SETQ ANCHORED (pop CONFIG))
                elseif (ILEQ TOPEND START)
                  then (RETURN NIL)                                  ; No more, failed
                else (add TOPEND -1)                                 ; First time or outer advance
                     (SETQ END TOPEND)
                     (SETQ TARGETTAIL TARGETLIST)
                     (SETQ HITEND NIL)
                     (SETQ ANCHORED NIL))
              (SETQ TARGET (CAR TARGETTAIL))
              (SELECTQ TARGET
                  (%# (CL:UNLESS (CDR TARGETTAIL)
                          (RETURN (LIST END (OR HITEND END))))
                      (CL:WHEN (ILEQ START END)                      ; If we are unanchored, slipping continues
                          (push STACK (LIST (SUB1 END)
                                            (CDR TARGETTAIL)
                                            (OR HITEND (SUB1 END))
                                            ANCHORED))))
                  (*  ;; Unanchored config for the tail that starts here.
                      (push STACK (LIST END (CDR TARGETTAIL)
                                        HITEND NIL)))
                  (if (SETQ RESULT (\TEDIT.BASICFIND.BACKWARD TSTREAM TARGET START END ANCHORED))
                      then (CL:UNLESS (CDR TARGETTAIL)               ; Success!
                               [RETURN (LIST (CAR RESULT)
                                             (OR HITEND (CADR RESULT])
                           (SETQ END (SUB1 (CADR RESULT)))           ; Next target
                           (CL:WHEN (ILEQ START END)
                               [push STACK (LIST END (CDR TARGETTAIL)
                                                 (OR HITEND (CADR RESULT])
                    elseif (NOT ANCHORED)
                      then (RETURN NIL])])
```

## (\TEDIT.BASICFIND.BACKWARD

```
  [LAMBDA (TSTREAM TARGETSTRING START END ANCHORED)                  ; Edited 23-Jun-2024 11:32 by rmk
                                                                     ; Edited 19-May-2024 23:07 by rmk
                                                                     ; Edited 17-Mar-2024 12:06 by rmk
                                                                     ; Edited 20-Jun-2023 00:11 by rmk
                                                                     ; Edited 30-May-91 20:56 by jds
```

;; Seach backwards thru TSTREAM for an exact match of TARGETSTRING.

;; Returns a (startmatch endmatch) pair of character positions in TSTREAM

```
    (bind LASTANCHOR (NCHARS _ (NCHARS TARGETSTRING))
          (ANCHOR _ (ADD1 END)) first (CL:WHEN (ZEROP NCHARS)
                                               (RETURN NIL))
                                      (CL:WHEN ANCHORED
                                          (SETQ START (IDIFFERENCE ANCHOR NCHARS)))
                                      ;; LASTANCHOR protects agains the beginning of the stream
                                      [SETQ LASTANCHOR (SUB1 (CL:IF ANCHORED
                                                                    ANCHOR
                                                                    (IPLUS START NCHARS))]
       eachtime (CL:WHEN (ILESSP ANCHOR LASTANCHOR)                  ; Won't fit in the frame
                     (RETURN NIL))
              (add ANCHOR -1)                                        ; Move the anchor back 1
              (\TEDIT.TEXTSETFILEPTR TSTREAM ANCHOR)
       when (for I from 1 do (CL:UNLESS (EQ (NTHCHARCODE TARGETSTRING I)
                                            (\TEDIT.TEXTBACKFILEPTR TSTREAM))
                                 (RETURN NIL))
                             (CL:WHEN (EQ I NCHARS)                  ; Matched the last char
                                 (RETURN T)))
       do (RETURN (LIST (IDIFFERENCE (ADD1 ANCHOR)
                                     NCHARS)
                        ANCHOR])
```

## (\TEDIT.PARSE.SEARCHSTRING

```
  [LAMBDA (TARGETSTRING BACKWARD)                                    ; Edited 23-Jun-2024 08:02 by rmk
                                                                     ; Edited 19-May-2024 22:43 by rmk
                                                                     ; Edited 19-Jun-2023 16:42 by rmk
                                                                     (* jds "31-Jan-84 13:26")
```

;; Parse TARGETSTRING into string-segments that are separated by the wild-card characters # and * (or escape). Each # is left as its own ;; segment, multiple *'s collapse to one, and *'s on the edges are removed. ' quotes the following character.

;; If BACKWARD, the search string segments are reverse, and the characters within each segment are reversed, so that the search can go ;; backwards.

;;

```
    (for CTAIL C SEGCODES on (CHCON TARGETSTRING) eachtime (SETQ C (CAR CTAIL))
       do (SELCHARQ C
```

```
                ((* ESCAPE)                                          ; Throw away the  first and multiiple *'s
                     (CL:WHEN SEGCODES
                         [push $$VAL (CONCATCODES (CL:IF BACKWARD
                                                  SEGCODES
                                                  (DREVERSE SEGCODES))]
                         (SETQ SEGCODES NIL))
                     (CL:WHEN (AND $$VAL (NEQ '* (CAR $$VAL)))
                         (push $$VAL '*)))
                (%#                                                  ; # stands alone
                     (CL:WHEN SEGCODES
                         [push $$VAL (CONCATCODES (CL:IF BACKWARD
                                                  SEGCODES
                                                  (DREVERSE SEGCODES))])
                     (push $$VAL '%#)
                     (SETQ SEGCODES NIL))
                (%'                                                  ; Quote the next character
                     (CL:WHEN (CDR CTAIL)
                         (push SEGCODES (CADR CTAIL))
                         (SETQ CTAIL (CDR CTAIL))))
                (push SEGCODES C))
        finally (if SEGCODES
                    then [push $$VAL (CONCATCODES (CL:IF BACKWARD
                                                  SEGCODES
                                                  (DREVERSE SEGCODES))]
                  elseif (EQ '* (CAR $$VAL))
                    then ;; Strip the first edge *
                         (pop $$VAL))
                (RETURN (CL:IF BACKWARD
                            $$VAL
                            (DREVERSE $$VAL))])

)
```

## FUNCTION INDEX