

```
'set novalue on' /* force KEXX and its way of SIGNAL ON NOVALUE */
```

```
/* Usage:      [MACRO] EURO                                     */
/* Example:    EURO                                           (finds next non-ASCII) */
/*            DEFine C-E EURO                               (maybe define key C-E) */
/* Purpose:    CLocate the next char. in the file area below */
/*            9: TAB, between 9 and 32: SP, or above 126: '~' */
/*            including 127: DEL and any 8-bit character.    */
/* Caveats:    10: LF and 13: CR are (intentionally) CLocated. */
/* Update 2008: KeditW 1.0 'cursor column' moves the cursor to */
/*            the FOCUS line, as the manual says. Apparently */
/*            Kedit 5 'cursor column' moves the cursor to the */
/*            CURRENT line after Locate followed by CLocate. */
/*            An added SOS MAKECURR fixed this Kedit oddity. */
/* History:    "PC-multilingual-850+euro" is the long name for */
/*            codepage 858. Of course this macro works with */
/*            any codepage containing US-ASCII as subset, but */
/*            not EBCDIC, UTF-1, UTF-7, UTF-16, or UTF-32.    */
/* See also:    KHELP THIGHlight, KHELP CLocate, KHELP Locate, */
/*            KHELP CURSOR, KHELP SOS MAKECURR                */
/* Requires:    Kedit 5.0 or KeditW 1.0 (Frank Ellermann, 2002) */
```

```
C =      xrange( '00'x, '08'x )      ; 'extract /WRAP/STAY'
C = C || xrange( '0A'x, '1F'x )      ; 'wrap on'
C = C || xrange( '7F'x, 'FF'x )      ; 'stay on'
```

```
if modifiable() <= command() + inprefix() then 'cursor column'
LINE = cursor.3()      ; L = LINE
POS  = cursor.4()      ; CL = POS      ; Z = 0
```

```
do N = 1 to length( C )
  Y = delimit( substr( C, N, 1 ) ) ; 'nomsg clocate' Y
  X = rc                                ; if X = 2 then iterate
  if X <> 0 then do
    'emsg FATAL' lastmsg.1() Y      ; exit X
  end
  Y = target.1() ; X = target.2() ; Z = 1
  'L : ' || LINE ; 'cl : ' || POS /* go back to start point */
  'refresh'      ; say 'found' c2x( substr( C, N, 1 ) )
```

```
if Y = LINE & X = POS then iterate
if L > LINE | ( L = LINE & CL > POS ) then do
  if Y = L & X > CL then iterate
  if Y < LINE | L < Y then iterate
  if Y = LINE & X < POS then iterate
end
if L < LINE | ( L = LINE & CL < POS ) then do
  if Y < LINE & L < Y then iterate
  if Y < LINE & ( L = Y & CL < X ) then iterate
  if Y = LINE & X < POS then do
    if L < Y | ( L = Y & CL < X ) then iterate
  end
end
L = Y ; CL = X /* note new minimal target */
end
```

```
'wrap' WRAP.1 ; 'stay' STAY.1 ; 'refresh'
```

```

if Z then do                                /* locating minimal target */
  'L :' || L 'sos makecurr'                /* SOS MAKECURR for KeditW */
X = substr( curline.3(), CL, 1 )
if CL > 1 then do
  'cl :' || CL - 1 ; 'cl +' delimit( X )
end                                          /* this +/- nonsense tries */
else do                                    /* to THIGHlight the char. */
  'cl :' || CL + 1 ; 'cl -' delimit( X )
end
say "" || c2x( X ) || "'x"                ; exit 0
end
'emsg all us-ascii (9 or 32..126)' ; exit 1

```