

File created: 17-Feb-2025 12:25:49 {WMEDLEY}<library>tedit>TEDIT-COMMAND.;135

edit by: rmk

changes to: (FNS \TEDIT.COMMAND.LOOP)

previous date: 28-Nov-2024 10:03:03 {WMEDLEY}<library>tedit>TEDIT-COMMAND.;133

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **TEDIT-COMMANDCOMS**

```
[[DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (\SCRATCHLEN 64))
  (CONSTANTS (MSPACE 153)
    (NSPACE 152)
    (THINSPACE 159)
    (FIGSPACE 154))
  (EXPORT (CONSTANTS (NONE.TTC 0)
    (CHARDELETE.TTC 1)
    (WORDDELETE.TTC 2)
    (DELETE.TTC 3)
    (FUNCTIONCALL.TTC 4)
    (REDO.TTC 5)
    (UNDO.TTC 6)
    (CMD.TTC 7)
    (NEXT.TTC 8)
    (EXPAND.TTC 9)
    (CHARDELETE.FORWARD.TTC 10)
    (WORDDELETE.FORWARD.TTC 11)
    (PUNCT.TTC 20)
    (TEXT.TTC 21)
    (WHITESPACE.TTC 22))
  (MACROS \TEDIT.MOUSESTATE \TEDIT.CHECK)
  (RECORDS TEDITTERMCODE)
  ;; Bits in the CHARTABLE that control line breaking, and what happens when a line is broken on this character (RMK:
  ;; THESE DON'T SEEM TO BE USED)
  (CONSTANTS (NOTBEFORE.LB 1)
    ; Must not break before this character (e.g. Japanese
    ; right-paren)
    (NOTAFTER.LB 2)
    ; Must not break after this character (e.g. Japanese open-quote)
    (BEFORE.LB 4)
    ; OK to break before this character, provided it's OK to break
    ; after the prior char (true of most non-white-space)
    (AFTER.LB 8)
    ; OK to break after this char, if it's OK to break before the next
    ; one (true of most white space)
    (DISAPPEAR-IF-NOT-SPLIT.LB 16)
    ; This character shouldn't be rendered if it isn't the last char on
    ; the line (non-breaking hyphen has this)
    (NEWCHAR-IF-SPLIT.LB 32)
    ; Look this char up in *TEDIT-SPLITCHAR-HASH* if this IS the
    ; last character on a line, and render it as the char we found.
  ]
(FNS \TEDIT.INTERRUPT.SETUP \TEDIT.MARKACTIVE \TEDIT.MARKINACTIVE \TEDIT.COMMAND.LOOP
  \TEDIT.COMMAND.RESET.SETUP)
[INITVARS (TEDIT.INTERRUPTS ' ((2 BREAK)
  (5 ERROR)
  (7 HELP)
  (20 CONTROL-T]

(VARS (| | NIL))
(GLOBALVARS | | TEDIT.INTERRUPTS)
(COMS
  (FNS \TEDIT.READTABLE \TEDIT.WORDBOUND.READTABLE TEDIT.GETSYNTAX TEDIT.SETSYNTAX TEDIT.GETFUNCTION
    TEDIT.SETFUNCTION TEDIT.WORDGET TEDIT.WORDSET TEDIT.ATOMBOUND.READTABLE)
  [DECLARE%: DONTEVAL@LOAD DOCOPY (VARS (TEDIT.READTABLE (\TEDIT.READTABLE))
    (TEDIT.WORDBOUND.READTABLE (\TEDIT.WORDBOUND.READTABLE])
  (GLOBALVARS TEDIT.READTABLE TEDIT.WORDBOUND.READTABLE))
[COMS
  (FILES (SYSLOAD FROM LISPUSERS)
    WHEELSCROLL)
  (FNS \TEDIT.WHEELSCROLL)
  (GLOBALVARS WHEELSCROLLCHARCODES)
  (VARS (WHEELSCROLLCHARCODES (\TEDIT.WHEELSCROLL])
  (COMS
    ; Clipboard
    (FNS \TEDIT.CLIPBOARD \TEDIT.COPYTOCLIPBOARD \TEDIT.EXTRACTOCLIPBOARD \TEDIT.WRITE.SEL)
    [DECLARE%: EVAL@LOAD EVAL@COMPILE DONTCOPY (CONSTANTS (CLIPBOARDCODES (CHARCODE (meta,C meta,X
      meta,c
      meta,X]
    (P (\TEDIT.CLIPBOARD])
(DECLARE%: EVAL@COMPILE DONTCOPY
(DECLARE%: EVAL@COMPILE
```

```
(RPAQQ \SCRATCHLEN 64)
(CONSTANTS (\SCRATCHLEN 64))
)
(DECLARE%: EVAL@COMPILE
(RPAQQ MSPACE 153)
(RPAQQ NSPACE 152)
(RPAQQ THINSPACE 159)
(RPAQQ FIGSPACE 154)
(CONSTANTS (MSPACE 153)
(NSPACE 152)
(THINSPACE 159)
(FIGSPACE 154))
)
```

:: FOLLOWING DEFINITIONS EXPORTED

```
(DECLARE%: EVAL@COMPILE
(RPAQQ NONE.TTC 0)
(RPAQQ CHARDELETE.TTC 1)
(RPAQQ WORDDELETE.TTC 2)
(RPAQQ DELETE.TTC 3)
(RPAQQ FUNCTIONCALL.TTC 4)
(RPAQQ REDO.TTC 5)
(RPAQQ UNDO.TTC 6)
(RPAQQ CMD.TTC 7)
(RPAQQ NEXT.TTC 8)
(RPAQQ EXPAND.TTC 9)
(RPAQQ CHARDELETE.FORWARD.TTC 10)
(RPAQQ WORDDELETE.FORWARD.TTC 11)
(RPAQQ PUNCT.TTC 20)
(RPAQQ TEXT.TTC 21)
(RPAQQ WHITESPACE.TTC 22)
(CONSTANTS (NONE.TTC 0)
(CHARDELETE.TTC 1)
(WORDDELETE.TTC 2)
(DELETE.TTC 3)
(FUNCTIONCALL.TTC 4)
(REDO.TTC 5)
(UNDO.TTC 6)
(CMD.TTC 7)
(NEXT.TTC 8)
(EXPAND.TTC 9)
(CHARDELETE.FORWARD.TTC 10)
(WORDDELETE.FORWARD.TTC 11)
(PUNCT.TTC 20)
(TEXT.TTC 21)
(WHITESPACE.TTC 22))
)
```

(DECLARE%: EVAL@COMPILE

(PUTPROPS \TEDIT.MOUSESTATE **MACRO** (BUTTON

:: Test to see if only the specified mouse button is down. DOES NOT call GETMOUSESTATE, so the
:: mouse-button info is the same as the last time it was called.

```
(SELECTQ (CAR BUTTON)
(LEFT ' (EQ LASTMOUSEBUTTONS 4))
(MIDDLE ' (EQ LASTMOUSEBUTTONS 1))
(RIGHT ' (EQ LASTMOUSEBUTTONS 2))
(SHOULDNT)))
```

```
(PUTPROPS \TEDIT.CHECK MACRO [ARGS (COND
[(AND (BOUNDP 'CHECK)
```

```

CHECK)
(CONS 'PROGN (for I in ARGS as J on ARGS when (NOT (STRINGP I))
collect (LIST 'OR I (LIST 'HELP "Tedit consistency-check
failure [RETURN to continue]: "
(COND
((STRINGP (CADR J)))
(T (KWOTE I]
(T (CONS COMMENTFLG ARGS])

```

```

)
(DECLARE%: EVAL@COMPILE
[ACCESSFNS TEDITTERMCODE ((TTCLASS (LOGAND DATUM 224))
(TTDECODE (LOGAND DATUM 31]
)

```

```

(DECLARE%: EVAL@COMPILE
(RPAQQ NOTBEFORE.LB 1)
(RPAQQ NOTAFTER.LB 2)
(RPAQQ BEFORE.LB 4)
(RPAQQ AFTER.LB 8)
(RPAQQ DISAPPEAR-IF-NOT-SPLIT.LB 16)
(RPAQQ NEWCHAR-IF-SPLIT.LB 32)

```

```

(CONSTANTS (NOTBEFORE.LB 1)
(NOTAFTER.LB 2)
(BEFORE.LB 4)
(AFTER.LB 8)
(DISAPPEAR-IF-NOT-SPLIT.LB 16)
(NEWCHAR-IF-SPLIT.LB 32))
)
)

```

:: END EXPORTED DEFINITIONS

(DEFINEQ

(\TEDIT.INTERRUPT.SETUP

```
[LAMBDA (PROC FORCEOFF)
```

```
; Edited 27-Mar-2024 15:27 by rmk
; Edited 22-Sep-2023 20:45 by rmk
(* jds "12-Sep-84 15:36")
```

:: Disarm any inconvenient interrupts, and save re-arming info on the window.

```

[LET ((TEXTOBJ (TEXTOBJ PROC T)))
(CL:WHEN TEXTOBJ
(UNINTERRUPTABLY
[COND
((AND FORCEOFF (PROCESSPROP PROC 'TEDIT.INTERRUPTS))
; There are disarmed interrupts; re-arm them.
(RESET.INTERRUPTS (PROCESSPROP PROC 'TEDIT.INTERRUPTS))
(PROCESSPROP PROC 'TEDIT.INTERRUPTS NIL))
([AND (NOT FORCEOFF)
(NOT (PROCESSPROP PROC 'TEDIT.INTERRUPTS))
; There aren't any interrupts disarmed; go do it.
(PROCESSPROP PROC 'TEDIT.INTERRUPTS (RESET.INTERRUPTS (OR (AND TEXTOBJ (GETTEXTPROP
TEXTOBJ
' INTERRUPTS))
TEDIT.INTERRUPTS)
T]))])
PROC])

```

(\TEDIT.MARKACTIVE

```
[LAMBDA (TEXTOBJ OPERATION)
```

```
; Edited 29-Jun-2024 10:32 by rmk
; Edited 12-Jun-90 18:04 by mitani
```

```
(replace (TEXTOBJ EDITOPACTIVE) of TEXTOBJ with OPERATION)
TEXTOBJ])
```

(\TEDIT.MARKINACTIVE

```
[LAMBDA (TEXTOBJ)
```

; Edited 12-Jun-90 18:04 by mitani

```
(replace (TEXTOBJ EDITOPACTIVE) of TEXTOBJ with NIL)
TEXTOBJ])
```

(\TEDIT.COMMAND.LOOP

```
[LAMBDA (TSTREAM)
```

```
; Edited 17-Feb-2025 12:05 by rmk
; Edited 28-Nov-2024 10:01 by rmk
; Edited 21-Nov-2024 11:51 by rmk
; Edited 13-Sep-2024 22:34 by rmk
; Edited 26-Aug-2024 23:26 by rmk
```



```

(\TEDIT.RESET.EXTEND.PENDING.DELETE TEXTOBJ)
(\TEDIT.SHOWSEL NIL T TEXTOBJ)))
(NEXT.TTC ; Move to the next blank to fill in, delimited by >>...<<
(TEDIT.NEXT TSTREAM))
(EXPAND.TTC ; EXPAND AN ABBREVIATION
(\TEDIT.ABBREV.EXPAND TSTREAM))
(SELECTC (AND CH (fetch TERMCLASS of (\SYNCODE (OR (FGETTOBJ TEXTOBJ TXTERMSA)
\PRIMTERMSA)
CH)))
(CHARDELETE.TC
(\TEDIT.CHARDELETE TSTREAM))
(WORDDELETE.TC
(\TEDIT.WORDDELETE TSTREAM))
(LINEDELETE.TC
(\TEDIT.DELETE TEXTOBJ))
(CL:WHEN CH ; Any other key: insert the character.
(\TEDIT.INSERT CH (TEXTSEL TEXTOBJ)
TSTREAM NIL T)))
(FSETTOBJ TEXTOBJ EDITOPACTIVE NIL)))
(FSETTOBJ TEXTOBJ EDITOPACTIVE NIL))]]

```

(\TEDIT.COMMAND.RESET.SETUP

[LAMBDA (ARGS STARTING)

```

; Edited 29-Jun-2024 00:10 by rmk
; Edited 17-Mar-2024 18:54 by rmk
; Edited 22-Feb-2024 23:14 by rmk
; Edited 5-Oct-2023 22:41 by rmk
; Edited 22-Sep-2023 20:41 by rmk
; Edited 16-Sep-2023 22:30 by rmk
; Edited 5-Nov-2022 10:41 by rmk
; Edited 12-Jun-90 18:04 by mitani

```

:: If STARTING is T, set up the reset-driven connections and values for editing; otherwise, break links and reset values for non-editing

```

(PROG ((TEXTOBJ (pop ARGS))
(OTTYWINDOW (pop ARGS))
(OTTYENTRYFN (pop ARGS))
(OTTYEXITFN (pop ARGS))
(OWINDOW (pop ARGS))
TTYWINDOW PRIMPANE)
(SETQ PRIMPANE (FGETTOBJ TEXTOBJ PRIMARYPANE))
[COND
(STARTING ; We're going INTO the command loop. Set up all the stuff
(FSETTOBJ TEXTOBJ EDITOPACTIVE T) ; Mark us busy until we're set up, so that nobody tries any funny
; stuff.

(SETQ OWINDOW (PROCESSPROP (THIS.PROCESS)
'WINDOW PRIMPANE)) ; Attach the process to this window.
(\TEDIT.INTERRUPT.SETUP (THIS.PROCESS)) ; Disarm all interrupt chars, re-arm them when we leave the edit
(SETQ OTTYEXITFN (PROCESSPROP (THIS.PROCESS)
'TTYEXITFN
'\TEDIT.PROCEXITFN)) ; Set up functions for getting in and out of the edit process
(SETQ OTTYENTRYFN (PROCESSPROP (THIS.PROCESS)
'TTYENTRYFN
'\TEDIT.PROCENTRYFN))
(CL:UNLESS (EQ (GETTEXTPROP TEXTOBJ 'TTYWINDOW)
'DON'T) ; He can suppress the ability to copy-select things into this
; window if he wants....

(SETQ TTYWINDOW (OR (GETTEXTPROP TEXTOBJ 'TTYWINDOW)
(CREATEW DEFAULTTTYREGION "TTY Window for TEdit" NIL T)))
(SETQ OTTYWINDOW (TTYDISPLAYSTREAM TTYWINDOW))
(PROCESSPROP (THIS.PROCESS)
'TEDITTTYWINDOW TTYWINDOW)
(WINDOWPROP TTYWINDOW 'PROCESS NIL)
[WINDOWPROP TTYWINDOW 'CLOSEFN (FUNCTION (LAMBDA (WW)
(WINDOWPROP WW 'PROCESS NIL)
; So that there isn't a circularity in the PROCESS ->
; TTYWINDOW -> PROCESS
(WINDOWPROP TTYWINDOW 'COPYINSERTFN (FUNCTION \TEDIT.COPYINSERTFN))
(WINDOWPROP TTYWINDOW 'MAINWINDOW PRIMPANE))
(FSETTOBJ TEXTOBJ TXTEDITING T) ; Tell TEdit that this document is actively being edited.
; Mark us un-busy so life can go on.

(FSETTOBJ TEXTOBJ EDITOPACTIVE NIL))

(T ; Coming OUT OF the command loop -- reset everything
(PROCESSPROP (THIS.PROCESS)
'WINDOW PRIMPANE) ; Detach the window from the edit process, to prevent circularity
; there

(WINDOWPROP PRIMPANE 'PROCESS NIL)
(\TEDIT.INTERRUPT.SETUP (THIS.PROCESS)
T) ; Re-arm the interrupts we turned off coming in.

(CL:WHEN (AND (TXTFILE TEXTOBJ)
(NOT (fetch (TEXTWINDOW CLOSINGFILE) of PRIMPANE)))
; Remember to close the file we were editing (Only if the window
; function isn't closing it.)
; Let anyone else who wants to close the file.
(CLOSEF? (TXTFILE TEXTOBJ))
(replace (TEXTWINDOW CLOSINGFILE) of PRIMPANE with NIL))
(PROCESSPROP (THIS.PROCESS)
'TTYEXITFN OTTYEXITFN)
(PROCESSPROP (THIS.PROCESS)

```

```

      'TTYENTRYFN OTTYENTRYFN)
(FSETTOBJ TEXTOBJ TXTHISTORY NIL) ; To prevent circularities arising from the need to remember
; textobjs in the history list.

(FSETTOBJ TEXTOBJ SELPANE NIL)
(FSETTOBJ TEXTOBJ TXTEDITING NIL) ; Tell TEdit that this document is NO LONGER actively being
; edited.

(CL:UNLESS (EQ (GETTEXTPROP TEXTOBJ 'TTYWINDOW)
              'DON'T)
           ; He can suppress the ability to copy-select things into this
           ; window if he wants....

      (TTYDISPLAYSTREAM OTTYWINDOW)
      (PROCESSPROP (THIS.PROCESS)
                   'TEDITTTYWINDOW NIL)) ]
(RETURN (LIST TEXTOBJ OTTYWINDOW OTTYENTRYFN OTTYEXITFN OWINDOW])
)

```

```

(RPAQ? TEDIT.INTERRUPTS ' ((2 BREAK)
                          (5 ERROR)
                          (7 HELP)
                          (20 CONTROL-T)))

```

```
(RPAQQ || NIL)
```

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

```
(GLOBALVARS || TEDIT.INTERRUPTS)
)
```

:: Read-table Utilities

```
(DEFINEQ
```

(\TEDIT.READTABLE

```
[LAMBDA NIL
```

; Edited 24-Dec-2023 09:54 by rmk
; Edited 20-Apr-2018 07:59 by rmk:
(* jds "12-Sep-86 13:48")

:: Create a TEdit read-table, to control which characters have what functions and call which commands.

```

(LET [(RTBL (create READTABLEP
                   READMACRODEFS _ (HASHARRAY 50)
;; CHARDELETE.FORWARD replaces WORDDELETE on ^W
      (for CH in (CHARCODE (BS ^A ^W DEL %#A %#B %#C ESC)) as CL
        in (CONSTANT (LIST CHARDELETE.TTC CHARDELETE.TTC CHARDELETE.FORWARD.TTC DELETE.TTC UNDO.TTC NEXT.TTC
                          CMD.TTC REDO.TTC))
          do
            (\SETSYNCODE (fetch READSA of RTBL)
                          CH CL)
            ; Set up the default syntax classes for command characters
      (for CH in (CHARCODE (^X)) as FN in ' (\TEDIT.ABBREV.EXPAND) do
        ; Set up the default function-calling characters (^X to expand
        ; abbrevs for now)
        (TEDIT.SETFUNCTION CH FN RTBL))

      (TEDIT.SETFUNCTION (CHARCODE ^O)
                        (FUNCTION GET.OBJ.FROM.USER)
                        RTBL)
        ; And for image object capture
      RTBL])
)

```

(\TEDIT.WORDBOUND.READTABLE

```
[LAMBDA NIL
```

; Edited 22-May-92 15:10 by jds

:: Create a readtable which will let TEdit find word boundaries. A word boundary is any point where the SYNCODE of the adjacent characters is
;; different

```

(PROG [(RTBL (create READTABLEP
                   READMACRODEFS _ (HARRAY 50)
      (for CH from 0 to 255 do (\SETSYNCODE (fetch READSA of RTBL)
                                           CH PUNCT.TTC))
;; By default, every character except those noted below is a punctuation character
      (for CH from (CHARCODE A) to (CHARCODE Z) do (\SETSYNCODE (fetch READSA of RTBL)
                                                                CH TEXT.TTC))
        ; Upper case alpha
      (for CH from (CHARCODE a) to (CHARCODE z) do (\SETSYNCODE (fetch READSA of RTBL)
                                                                CH TEXT.TTC))
        ; Lower case alpha
      (for CH from (CHARCODE 0) to (CHARCODE 9) do (\SETSYNCODE (fetch READSA of RTBL)
                                                                CH TEXT.TTC))
        ; And digits are text characters
;; European chars and accents are text characters:
      (for CH from (CHARCODE "361,41") to (CHARCODE "361,376") do (\SETSYNCODE (fetch READSA of RTBL)
                                                                                CH TEXT.TTC))
      (for CH from (CHARCODE "0,301") to (CHARCODE "0,317") do (\SETSYNCODE (fetch READSA of RTBL)
                                                                                CH TEXT.TTC))
      (for CH from (CHARCODE "0,341") to (CHARCODE "0,376") do (\SETSYNCODE (fetch READSA of RTBL)
                                                                                CH TEXT.TTC))
      (for CH in (CHARCODE (CR SPACE TAB ^L)) do (\SETSYNCODE (fetch READSA of RTBL)

```

```

                                CH WHITESPACE.TTC))
                                ; And these are white space
    (for CH in (LIST MSPACE NSPACE THINSPACE FIGSPACE) do (\SETSYNCODE (fetch READSA of RTBL)
                                CH TEXT.TTC))
    (RETURN RTBL])

```

(TEDIT.GETSYNTAX

[LAMBDA (CH TABLE)

; Edited 24-Dec-2023 09:47 by rmk
; Edited 31-Mar-87 10:01 by jds
; Find TEdit's interpretation of a given character

```

(SELECTC (\SYNCODE [fetch READSA of (COND
    ((type? TEXTOBJ TABLE) ; If given a TEXTOBJ in place of a read table, coerce it to the
    ; read table for that edit session
    (OR (fetch (TEXTOBJ TXTRTBL) of TABLE)
        TEDIT.READTABLE))
    ((type? STREAM TABLE) ; If given a TEXTOBJ in place of a read table, coerce it to the
    ; read table for that edit session
    (OR (fetch (TEXTOBJ TXTRTBL) of (fetch (TEXTSTREAM TEXTOBJ) of TABLE))
        TEDIT.READTABLE))
    (T (OR TABLE TEDIT.READTABLE]
    (COND
        ((LITATOM CH) ; Symbols are converted to numeric charcodes
         (APPLY* 'CHARCODE CH))
        ((STRINGP CH) ; As are string char-names
         (APPLY* 'CHARCODE CH))
        (T CH)))
    (WORDDELETE.TTC
     'WORDDELETE)
    (WORDDELETE.FORWARD.TTC
     'WORDDELETE.FORWARD)
    (CHARDELETE.TTC
     'CHARDELETE)
    (CHARDELETE.FORWARD.TTC
     'CHARDELETE.FORWARD)
    (DELETE.TTC 'DELETE)
    (UNDO.TTC 'UNDO)
    (REDO.TTC 'REDO)
    (FUNCTIONCALL.TTC
     'FN)
    (CMD.TTC 'CMD)
    (NEXT.TTC 'NEXT)
    (EXPAND.TTC 'EXPAND)
    NIL])

```

(TEDIT.SETSYNTAX

[LAMBDA (CHAR CLASS TABLE)

; Edited 24-Dec-2023 09:17 by rmk
; Edited 31-Mar-87 10:00 by jds
; SETS TEDIT-STYLE SYNTAX BITS IN A TERMTABLE

```

(PROG1 (TEDIT.GETSYNTAX (SETQ CHAR (COND
    ((LITATOM CHAR)
     (APPLY* 'CHARCODE CHAR))
    ((STRINGP CHAR)
     (APPLY* 'CHARCODE CHAR))
    (T CHAR)))
    TABLE)
    (\SETSYNCODE [fetch READSA of (COND
    ((type? TEXTOBJ TABLE) ; If given a TEXTOBJ in place of a read table, coerce it to the
    ; read table for that edit session
    (OR (fetch (TEXTOBJ TXTRTBL) of TABLE)
        TEDIT.READTABLE))
    ((type? STREAM TABLE) ; If given a TEXTOBJ in place of a read table, coerce it to the
    ; read table for that edit session
    (OR (fetch (TEXTOBJ TXTRTBL) of (fetch (TEXTSTREAM TEXTOBJ) of TABLE))
        TEDIT.READTABLE))
    (T (OR TABLE TEDIT.READTABLE]
    CHAR
    (SELECTQ CLASS
        (CHARDELETE CHARDELETE.TTC)
        (CHARDELETE.FORWARD
         CHARDELETE.FORWARD.TTC)
        (WORDDELETE WORDDELETE.TTC)
        (WORDDELETE.FORWARD
         WORDDELETE.FORWARD.TTC)
        ((DELETE LINEDELETE)
         DELETE.TTC)
        (UNDO UNDO.TTC)
        (REDO REDO.TTC)
        (CMD CMD.TTC)
        (FN FUNCTIONCALL.TTC)
        (NEXT NEXT.TTC)
        (EXPAND EXPAND.TTC)
        NONE.TTC)))])

```

(TEDIT.GETFUNCTION

```
[LAMBDA (CHARCODE TABLE)
  (SETQ TABLE (COND
    ((type? TEXTOBJ TABLE)
      (* If given a TEXTOBJ in place of a read table, coerce it to the read table for that edit session)
      (fetch (TEXTOBJ TXTRTBL) of TABLE))
    ((type? STREAM TABLE)
      (* If given a TEXTOBJ in place of a read table, coerce it to the read table for that edit session)
      (fetch (TEXTOBJ TXTRTBL) of (fetch (TEXTSTREAM TEXTOBJ) of TABLE)))
    (T (OR TABLE TEDIT.READTABLE]
  (SETQ CHARCODE (COND
    ((LITATOM CHARCODE)
      (APPLY* 'CHARCODE CHARCODE))
    (T CHARCODE)))
  (AND TABLE (type? READTABLEP TABLE)
    (IEQP FUNCTIONCALL.TTC (\SYNCODE (fetch READSA of TABLE)
      CHARCODE))
    (fetch READMACRODEFS of TABLE)
    (CAR (FETCH MACROFN OF (GETHASH CHARCODE (fetch READMACRODEFS of TABLE)))))
```

(* jds "19-Sep-85 17:06")
 (* Gets the FN that is called when CH is hit inside TEDIT.)

; Edited 31-Mar-87 10:58 by jds
 ; Set TEDITs (read) table so that FN is called whenever
 ; CHARCODE is typed.
 ; If FN is NIL, make the character be normal again.

(TEDIT.SETFUNCTION

```
[LAMBDA (CHARCODE FN RTBL)
  (SETQ RTBL (COND
    ((type? TEXTOBJ RTBL)
      (fetch (TEXTOBJ TXTRTBL) of RTBL))
    ((type? STREAM RTBL)
      (fetch (TEXTOBJ TXTRTBL) of (fetch (TEXTSTREAM TEXTOBJ) of RTBL)))
    (T (OR RTBL TEDIT.READTABLE]
  (\SETSYNCODE (fetch READSA of RTBL)
    (SETQ CHARCODE (COND
      ((LITATOM CHARCODE)
        (APPLY* 'CHARCODE CHARCODE))
      ((STRINGP CHARCODE)
        (APPLY* 'CHARCODE CHARCODE))
      (T CHARCODE)))
    (COND
      (FN
        FUNCTIONCALL.TTC)
      (T
        NONE.TTC)))
  (OR (fetch READMACRODEFS of RTBL)
    (replace READMACRODEFS of RTBL with (HARRAY 50)))
  (PUTHASH CHARCODE (CREATE READMACRODEF
    MACROTYPE _ 'TEDIT
    MACROFN _ (LIST FN))
    (fetch READMACRODEFS of RTBL])
```

; If given a TEXTOBJ in place of a read table, coerce it to the
 ; read table for that edit session

; If given a TEXTOBJ in place of a read table, coerce it to the
 ; read table for that edit session

; He gave us a function to call. Set up the syntax so it IS called.

; He gave us a function of NIL, meaning 'turn it off' . Cause this
 ; character to become normal.
 ; Mark the character as invoking a function

; Make sure there's a hash table to store the function in.

(TEDIT.WORDGET

```
[LAMBDA (CH TABLE)
  (\SYNCODE (fetch READSA of (OR TABLE TEDIT.WORDBOUND.READTABLE))
  (COND
    ((SMALLP CH))
    (T (CHCON1 CH]))
```

(* jds "27-MAY-83 13:24")

(TEDIT.WORDSET

```
[LAMBDA (CHARCODE CLASS TABLE)
  (\SETSYNCODE (fetch READSA of (OR TABLE TEDIT.WORDBOUND.READTABLE))
  (COND
    ((SMALLP CHARCODE))
    (T (CHCON1 CHARCODE)))
  (COND
    ((FIXP CLASS))
    (T (SELECTQ CLASS
      (PUNCTUATION PUNCT.TTC)
      (WHITESPACE WHITESPACE.TTC)
      (TEXT TEXT.TTC)
      TEXT.TTC]))
```

(* jds " 1-JUN-83 12:23")
 (* SETS TEDIT-STYLE SYNTAX BITS IN A TERMTABLE)

(TEDIT.ATOMBOUND.READTABLE

```
[LAMBDA (READTABLE)
  ;; A wordbound table that approximates the unquoted OTHER characters of Lisp atoms as defined by READTABLE or the current readtable. This
```

; Edited 25-Dec-2023 13:10 by rmk
 ; Edited 5-Dec-2023 23:47 by rmk

;; is specified as the BOUNDTABLE for Lisp source code edits. Not perfect, but not bad.

;; Could cache this for common readtables (interlisp, commonlisp)

```
(LET ((TABLE (\TEDIT.WORDBOUND.READTABLE))) ;\TEDIT.WORDBOUND.READTABLE creates a new one each
      ;time.
      (for CODE IN (GETSYNTAX 'OTHER (OR READTABLE *READTABLE*)) do (TEDIT.WORDSET CODE 'TEXT TABLE))
      (for CODE IN (GETSYNTAX 'BREAK (OR READTABLE *READTABLE*)) do (TEDIT.WORDSET CODE 'PUNCTUATION TABLE))
      (TEDIT.WORDSET (CHARCODE %:)
                    'TEXT TABLE)
      TABLE])
```

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQ TEDIT.READTABLE (\TEDIT.READTABLE))

(RPAQ TEDIT.WORDBOUND.READTABLE (\TEDIT.WORDBOUND.READTABLE))

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.READTABLE TEDIT.WORDBOUND.READTABLE)

)

;; Wheelscroll

(FILESLOAD (SYSLOAD FROM LISPUSERS)
 WHEELSCROLL)

(DEFINEQ

(\TEDIT.WHEELSCROLL

[LAMBDA NIL

; Edited 2-Oct-2023 23:23 by rmk

;; TEDIT disables interrupts, so it has to deal with wheelscroll behaviors when the caret is in the Tedit window. Each of the individual actions is
;; conditioned on WHEELSCROLLENABLED (which may or may not have been loaded).

;; This localizes the behavior inside Tedit, where we also suppress Tedit from thinking that somehow these characters change the selection
;; highlighting.

```
(for I in WHEELSCROLLINTERRUPTS collect (TEDIT.SETFUNCTION (CAR I)
                  '[LAMBDA NIL
                    (AND WHEELSCROLLENABLED , (CADR I)]
                  TEDIT.READTABLE)
      (CAR I]))
```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS WHEELSCROLLCHARCODES)

)

(RPAQ WHEELSCROLLCHARCODES (\TEDIT.WHEELSCROLL))

;; Clipboard

(DEFINEQ

(\TEDIT.CLIPBOARD

[LAMBDA NIL

; Edited 21-Apr-2024 09:57 by rmk

; Edited 2-Oct-2023 23:23 by rmk

;; TEDIT disables interrupts, so it has to deal with special interrupt behaviors when the caret is in the Tedit window. This localizes the behavior of
;; WHEELSCROLL and CLIPBOARD inside Tedit.

;; Clipboard paste

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,v")
  (FUNCTION PASTEFROMCLIPBOARD)
  TEDIT.READTABLE)
```

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,V")
  (FUNCTION PASTEFROMCLIPBOARD)
  TEDIT.READTABLE)
```

;; Clipboard copy

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,c")
  (FUNCTION \TEDIT.COPYTOCLIPBOARD)
  TEDIT.READTABLE)
```

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,C")
  (FUNCTION \TEDIT.COPYTOCLIPBOARD)
  TEDIT.READTABLE)
```

;; Clipboard extract

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,X")
  (FUNCTION \TEDIT.EXTRACTTOCLIPBOARD)
  TEDIT.READTABLE)
```

```
(TEDIT.SETFUNCTION (CHARCODE "Meta,x")
```

```
(FUNCTION \TEDIT.EXTRACTTOCLIPBOARD)
TEDIT.READTABLE)
```

:: Each of the individual actions is conditioned on WHEELSCROLLENABLED (which may or may not have been loaded).

```
(for I in WHEELSCROLLINTERRUPTS collect (TEDIT.SETFUNCTION (CAR I)
'[LAMBDA NIL
(AND WHEELSCROLLENABLED , (CADR I])
TEDIT.READTABLE)
(CAR I]))
```

(\TEDIT.COPYTOCLIPBOARD

```
[LAMBDA (TSTREAM TEXTOBJ SEL EXTRACT) ; Edited 21-Apr-2024 11:51 by rmk
; Edited 2-Apr-2024 17:01 by rmk
; Edited 18-Apr-2018 00:02 by rmk:
```

:: If CLIPBOARD is loaded, this copies the characters in the current selection to the clipboard (SEL argument is ignored).

```
(CL:WHEN (FGETD (FUNCTION PUTCLIPBOARD))
(SETQ TSTREAM (TEXTSTREAM (OR TSTREAM (TTY.PROCESS))
T))
(CL:WHEN TSTREAM
(PUTCLIPBOARD TSTREAM (FUNCTION \TEDIT.WRITE.SEL))
(CL:WHEN EXTRACT (TEDIT.DELETE TSTREAM))))])
```

(\TEDIT.EXTRACTTOCLIPBOARD

```
[LAMBDA (TSTREAM TEXTOBJ SEL) ; Edited 21-Apr-2024 09:20 by rmk
(\TEDIT.COPYTOCLIPBOARD TSTREAM TEXTOBJ SEL T])
```

(\TEDIT.WRITE.SEL

```
[LAMBDA (TSTREAM STREAM) ; Edited 21-Apr-2024 11:55 by rmk
```

:: Writes the selected characters in TSTREAM to STREAM.

:: If there are no image objects, this is equivalent to (PRIN3 (TEDIT.SEL.AS.STRING ...)), but that would trip over image objects. Image objects could be skipped, or as here, represented as the OBJECTBYTE or described in some way.

:: For Medley-to-Medley copy/paste we could also create a local tmp stream that shadows the system clipboard, and apply the PUTFN to that stream. Then copy/paste could be used to move image objects around with a single Medley or perhaps across Medley's (if the GETFN is available).

```
(LET* ((TEXTOBJ (TEXTOBJ TSTREAM))
(SEL (FGETTOBJ TEXTOBJ SEL)))
(CL:WHEN (IGREATERP (GETSEL SEL DCH)
0)
```

:: This could be run by setting the fileptr and doing BIN's. This way we don't manipulate TSTREAM's file position FWIW.

```
(for I CODE PRE (OBJECTBYTE _ (GETTEXTPROP TEXTOBJ 'OBJECTBYTE))
(NOJECTS _ 0) from (GETSEL SEL CH#) to (SUB1 (GETSEL SEL CHLIM))
while (SETQ CODE (TEDIT.NTHCHARCODE TSTREAM I))
do (if (CHARCODEP CODE)
then (PRINTCCODE CODE STREAM)
elseif (IMAGEOBJP CODE)
then (add NOJECTS 1)
(if OBJECTBYTE
then (PRINTCCODE OBJECTBYTE STREAM)
else (PRIN3 "{" STREAM)
(PRIN4 (IMAGEOBJPROP CODE 'GETFN)
STREAM)
(CL:WHEN (SETQ PRE (APPLY* (OR (IMAGEOBJPROP CODE 'PREPRINTFN)
(FUNCTION NIL))
PRE CODE))
(PRIN3 " : " STREAM)
(PRIN4 PRE STREAM))
(PRIN3 "}" STREAM))
else (ERROR "UNRECOGNIZED TEDIT CHARACTER" CODE))
finally (CL:WHEN (IGREATERP NOJECTS 0)
(TEDIT.PROMPTPRINT TSTREAM (CONCAT "Note: Selection contains " NOJECTS " image
object" (CL:IF (EQ NOJECTS 1)
""
"s"))
T))))))
```

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

```
(RPAQ CLIPBOARDCODES (CHARCODE (meta,C meta,X meta,c meta,X)))
```

```
[CONSTANTS (CLIPBOARDCODES (CHARCODE (meta,C meta,X meta,c meta,X)
)
)
```

(\TEDIT.CLIPBOARD)

FUNCTION INDEX

TEDIT.ATOMBOUND.READTABLE	8	\TEDIT.CLIPBOARD	9	\TEDIT.MARKINACTIVE	3
TEDIT.GETFUNCTION	7	\TEDIT.COMMAND.LOOP	3	\TEDIT.READTABLE	6
TEDIT.GETSYNTAX	7	\TEDIT.COMMAND.RESET.SETUP	5	\TEDIT.WHEELSCROLL	9
TEDIT.SETFUNCTION	8	\TEDIT.COPYTOCLIPBOARD	10	\TEDIT.WORDBOUND.READTABLE	6
TEDIT.SETSYNTAX	7	\TEDIT.EXTRACTTOCLIPBOARD	10	\TEDIT.WRITE.SEL	10
TEDIT.WORDGET	8	\TEDIT.INTERRUPT.SETUP	3		
TEDIT.WORDSET	8	\TEDIT.MARKACTIVE	3		

CONSTANT INDEX

AFTER.LB	3	FIGSPACE	2	PUNCT.TTC	2
BEFORE.LB	3	FUNCTIONCALL.TTC	2	REDO.TTC	2
CHARDELETE.FORWARD.TTC	2	MSPACE	2	TEXT.TTC	2
CHARDELETE.TTC	2	NEWCHAR-IF-SPLIT.LB	3	THINSPACE	2
CLIPBOARDCODES	10	NEXT.TTC	2	UNDO.TTC	2
CMD.TTC	2	NONE.TTC	2	WHITESPACE.TTC	2
DELETE.TTC	2	NOTAFTER.LB	3	WORDELETE.FORWARD.TTC	2
DISAPPEAR-IF-NOT-SPLIT.LB	3	NOTBEFORE.LB	3	WORDELETE.TTC	2
EXPAND.TTC	2	NSPACE	2	\SCRATCHLEN	2

VARIABLE INDEX

.....	6	TEDIT.READTABLE	9	WHEELSCROLLCHARCODES	9
TEDIT.INTERRUPTS	6	TEDIT.WORDBOUND.READTABLE	9		

MACRO INDEX

\TEDIT.CHECK	2	\TEDIT.MOUSESTATE	2
--------------------	---	-------------------------	---

RECORD INDEX

TEDITTERMCODE	3
---------------------	---
