

File created: 25-Feb-2025 15:07:03 {WMEDLEY}<library>TEDIT>TEDIT.;765

edit by: rmk

changes to: (FNS TEDIT.MAP.OBJECTS TEDIT.INSERT.OBJECT)

previous date: 20-Feb-2025 08:50:50 {WMEDLEY}<library>TEDIT>TEDIT.;763

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ TEDITCOMS

```
(FILES (SYSLOAD)
  POSTSCRIPTSTREAM PDFSTREAM)
(COMS                                     ; Loadup stuff
  (VARS TEDITFILES)
  (FNS MAKE-TEDIT-EXPORTS.ALL UPDATE-TEDIT EDIT-TEDIT)
  (DECLARE%: DONTEVAL@LOAD DONTCOPY DONTEVAL@COMPILE
    ;; This gets EXPORTS.ALL loaded when TEDIT-EXPORTS.ALL is loaded
    (EXPORT (FILES (FROM LOADUPS)
      EXPORTS.ALL))
  (DECLARE%: EVAL@COMPILE DONTCOPY (FILES TEDIT-EXPORTS.ALL))
  (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
    UNICODE)))
(DECLARE%: EVAL@COMPILE DONTCOPY
  ;; Assertions go to comments if not being checked, so we see value-warnings
  (EXPORT (COMS (MACROS TEDIT-ASSERT)
    (GLOBALVARS CHECK-TEDIT-ASSERTIONS)
    (INITVARS (CHECK-TEDIT-ASSERTIONS T)))
    (MACROS OBJECT.ALLOWS)))
(FILES TEDIT-PCTREE TEDIT-STREAM TEDIT-COMMAND TEDIT-SCREEN TEDIT-ABBREV TEDIT-LOOKS TEDIT-STYLES)
[VARS (TEDIT.STARTUP.MONITORLOCK (CREATE.MONITORLOCK 'TEDIT.STARTUP)
  (INITVARS (TEDIT.TENTATIVE NIL)
    (TEDIT.DEFAULT.PROPS NIL))
  (GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)
  ;; Unslashed functions. Public?
  (FNS TEDIT TEXTSTREAM TEXTSTREAMP TEDITMENUUP COERCETEXTSTREAM TEDIT.CONCAT TEDITSTRING TEDIT-SEE
    TEDIT.COPY TEDIT.DELETE TEDIT.INSERT TEDIT.TERPRI TEDIT.KILL TEDIT.QUIT TEDIT.MOVE
    TEDIT.STRINGWIDTH TEDIT.CHARWIDTH)
  (FNS TEXTOBJ COERCETEXTOBJ)
  (MACROS TEVAL)
  (FNS TDRIbble)
  (COMS                                     ; Object-oriented editing
    (FNS TEDIT.INSERT.OBJECT TEDIT.EDIT.OBJECT TEDIT.FIND.OBJECT TEDIT.FIND.OBJECT.BACKWARD
      TEDIT.OBJECT.CHANGED TEDIT.MAP.OBJECTS \TEDIT.FIRST.OBJPIECE \TEDIT.NEXT.OBJPIECE)
    (FILES IMAGEOBJ))
  (FNS \TEDIT.CONCAT.PAGEFRAMES \TEDIT.GET.PAGE.HEADINGS \TEDIT.CONCAT.INSTALL.HEADINGS)
  (FNS \TEDIT.MOVE.MSG \TEDIT.READONLY)
  (FNS TEDIT.NCHARS TEDIT.RPLCHARCODE TEDIT.NTHCHARCODE TEDIT.NTHCHAR \TEDIT.PIECE.NTHCHARCODE)
  ;; Slashed functions. Private?
  (FNS \TEDIT1 \TEDIT.INSERT \TEDIT.MOVE \TEDIT.COPY \TEDIT.REPLACE.SELPIECES \TEDIT.INSERT.SELPIECES
    \TEDIT.RESTARTFN \TEDIT.CHARDELETE \TEDIT.CHARDELETE \TEDIT.COPYPIECE \TEDIT.APPLY.OBJFN \TEDIT.DELETE
    \TEDIT.DIFFUSE.PARALOOKS \TEDIT.WORDDELETE \TEDIT.WORDDELETE.FORWARD \TEDIT.FINISHEDIT?)
  (COMS (FNS \TEDIT.HELP)
    (INITVARS (\TEDIT.THELPFLG NIL)))
  (FNS \TEDIT.PARAPIECES \TEDIT.PARA.FIRST \TEDIT.PARA.LAST)
  (FNS \TEDIT.WORD.FIRST \TEDIT.WORD.LAST)
  ;; Would be nice to just do (DOFILESLOAD (CDR TEDITFILES)). But the order for exports.all and the order for loading have to be aligned.
  (FILES TEDIT-FIND TEDIT-HISTORY TEDIT-FILE TEDIT-OLDFILE TEDIT-WINDOW TEDIT-SELECTION TEDIT-TFBRavo
    TEDIT-HCPY TEDIT-PAGE TEDIT-BUTTONS TEDIT-MENU TEDIT-FNKEYS)
  (COMS                                     ; TEDIT Support information
    (E (SETQ TEDITSYSTEMDATE (DATE)))
    (VARS TEDITSYSTEMDATE))
  (COMS                                     ; LISTFILES Interface, so the system can decide if a file is a
    ; Tedit file.
    (ADDVARS (PRINTFILETYPES (TEDIT (TEST \TEDIT.GET.TRAILER)
      (EXTENSION (TEDIT))
    (FILESLOAD (SYSLOAD)
      POSTSCRIPTSTREAM PDFSTREAM)
```

;; Loadup stuff

```
(RPAQQ TEDITFILES (TEDIT TEDIT-PCTREE TEDIT-SELECTION TEDIT-SCREEN TEDIT-STREAM TEDIT-COMMAND TEDIT-FILE
  TEDIT-OLDFILE TEDIT-LOOKS TEDIT-STYLES TEDIT-WINDOW TEDIT-BUTTONS TEDIT-MENU TEDIT-FIND
  TEDIT-FNKEYS TEDIT-HCPY TEDIT-HISTORY TEDIT-PAGE TEDIT-ABBREV TEDIT-TFBRavo))
```

(DEFINEQ

(MAKE-TEDIT-EXPORTS.ALL

[LAMBDA NIL

; Edited 29-Aug-2023 21:59 by rmk
; Edited 11-Sep-2022 23:43 by rmk

```
(LET (VAL)
  [CNDIR (PROG1 (DIRECTORYNAME T)
    (CNDIR (MEDLEYDIR "library>tedit"))
    (SETQ VAL (GATHEREXPORTS TEDITFILES (MEDLEYDIR "library/tedit" "tedit-exports.all" T))))]
  VAL])
```

(UPDATE-TEDIT

[LAMBDA (FILES LDFLG)

; Edited 16-Feb-2025 11:25 by rmk
; Edited 26-Oct-2022 21:10 by rmk

```
(CL:UNLESS LDFLG (SETQ LDFLG T))
(for F in LOADEDFILELST eachtime (SETQ F (TRUEFILENAME F))
  when [AND (STRPOS ">library>tedit>TEDIT-" F 1 NIL NIL NIL UPPERARRAY)
    (STRING.EQUAL 'LCOM (FILENAMEFIELD F 'EXTENSION)]
  collect (FILENAMEFIELD F 'NAME) finally
```

;; Loading TEDIT will probably do the DOFILESLOAD for all the other files, this may be overkill. But
;; we want to make sure the load's happen even if it looks like the files are already there (e.g. not
;; LOAD?).

```
(RETURN (for LF in (CONS 'TEDIT (REMOVE 'TEDIT $VAL))
  collect (PSEUDOFILNAME (LOAD LF LDFLG))
```

(EDIT-TEDIT

[LAMBDA NIL

; Edited 3-Jul-2023 13:44 by rmk
; Edited 17-Jun-2023 10:00 by rmk
; Edited 25-Apr-2023 17:39 by rmk
; Edited 26-Oct-2022 21:12 by rmk
; Edited 14-Sep-2022 08:37 by rmk

```
(BKSYSEBUF " ")
(RESETLST
  (RESETSAVE LOADDDBFLG 'YES)
  (FOR F IN TEDITFILES DO (LOADFROM F)
    (LOADCOMP F)))
(% . ANALYZE ON IN TEDITFILES])
```

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

;; FOLLOWING DEFINITIONS EXPORTED

```
(FILESLOAD (FROM LOADUPS)
  EXPORTS.ALL)
)
```

;; END EXPORTED DEFINITIONS

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

```
(FILESLOAD TEDIT-EXPORTS.ALL)
)
```

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

```
(FILESLOAD (LOADCOMP)
  UNICODE)
)
```

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

;; FOLLOWING DEFINITIONS EXPORTED

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS TEDIT-ASSERT MACRO [ARGS (COND
  [CHECK-TEDIT-ASSERTIONS ` (CL:UNLESS , (CAR ARGS)
    [\bTEDIT.THELP "TEDIT-ASSERT FAILURE"
    , (KWOTE (CAR ARGS))]]
  (T ` (* (TEDIT-ASSERT (\,@ ARGS)))]))
)
```

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

```
(GLOBALVARS CHECK-TEDIT-ASSERTIONS)
)
```

```
(RPAQ? CHECK-TEDIT-ASSERTIONS T)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS OBJECT.ALLOWS MACRO ((PC OPERATION FROMTOBJ TOTOBJ)
  (OR (NOT (EQ OBJECT.PTYPE (PTYPE PC))))
```

(\TEDIT.APPLY.OBJFN (PCONTENTS PC)
OPERATION FROMTOBJ TOTOBJ)))

)
)

:: END EXPORTED DEFINITIONS

(FILESLOAD TEDIT-PCTREE TEDIT-STREAM TEDIT-COMMAND TEDIT-SCREEN TEDIT-ABBREV TEDIT-LOOKS TEDIT-STYLES)

(RPAQ TEDIT.STARTUP.MONITORLOCK (CREATE.MONITORLOCK 'TEDIT.STARTUP))

(RPAQ? TEDIT.TENTATIVE NIL)

(RPAQ? TEDIT.DEFAULT.PROPS NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)

)

:: Unslashed functions. Public?

(DEFINEQ

(TEDIT

[LAMBDA (TEXT WINDOW DONTSPAWN PROPS)

:: Edited 25-Jun-2024 11:59 by rmk

:: Edited 9-Mar-2024 22:47 by rmk

:: Edited 20-Oct-2023 11:02 by rmk

:: Edited 17-Oct-2023 08:53 by rmk

:: Edited 22-Jun-2022 20:01 by rmk: Call to OPENSTREAM passes FORMAT from PROPS

:: Edited 31-Jan-2022 17:19 by rmk: String TEXT is a file name

:: Edited 24-Dec-2021 19:21 by rmk

:: Edited 11-Jun-99 14:14 by rmk:

:: Edited 3-Jun-88 14:27 by jds

:: User entry to the text editor. Takes an optional window to be used for editing

(SETQ PROPS (APPEND PROPS TEDIT.DEFAULT.PROPS))

:: DONTSPAWN => Don't try to create a new process for this edit.

(LET (TSTREAM PROC)

(push PROPS 'BEING-EDITED T)

(SETQ TSTREAM (OPENTEXTSTREAM TEXT (OR WINDOW 'Tedit)
NIL NIL PROPS))

(SETQ WINDOW (\TEDIT.PRIMARYPANE TSTREAM))

(COND

(DONTSPAWN

(\TEDIT1 TSTREAM WINDOW T))

(T

[SETQ PROC (ADD.PROCESS (LIST (FUNCTION \TEDIT1)
(KWOTE TSTREAM)
WINDOW NIL)

'NAME

'Tedit

'RESTARTABLE

'HARDRESET

'RESTARTFORM

(LIST (FUNCTION \TEDIT.RESTARTFN)

(KWOTE TSTREAM)

WINDOW

(KWOTE PROPS]

(PROCESSPROP PROC 'WINDOW WINDOW)

(CL:UNLESS (LISTGET PROPS 'LEAVETTY)

(TTY.PROCESS PROC))

PROC])

; Make sure the file exists before trying to open the window.
; Mark the document as actively in edit, so caret flashes when
; the window first opens.

; Either no processes running, or specifically not to spawn one.

; Spawn a process to do the edit.

; Unless he asked us to leave the tty where it is, TEdit should get
; it.

(TEXTSTREAM

[LAMBDA (TSTREAM? NOERROR)

; Edited 29-Apr-2024 12:50 by rmk
; Edited 20-Mar-2024 08:51 by rmk
; Edited 24-Mar-2023 18:01 by rmk
(* jds "11-Jul-85 12:06")

:: Convert from any designator of a textstream to that textstream.

(LET (TS WINDOW X)

(SETQ TS (if (type? TEXTSTREAM TSTREAM?)

then TSTREAM?

elseif (type? TEXTOBJ TSTREAM?)

then (FGETTOBJ TSTREAM? STREAMHINT)

elseif (SETQ WINDOW (if (WINDOW TSTREAM?)

```

elseif (PROCESSP TSTREAM?)
  then (PROCESS.WINDOW TSTREAM?)
elseif (DISPLAYSTREAMP TSTREAM?)
  then (WFROMDS TSTREAM?))
then (if (type? TEXTSTREAM (SETQ X (fetch (TEXTWINDOW WTEXTSTREAM) of WINDOW)))
  then X
  elseif (type? TEXTOBJ (SETQ X (fetch (TEXTWINDOW WTEXTOBJ) of WINDOW)))
    then (FGETTOBJ X STREAMHINT))
elseif (AND (type? SELECTION TSTREAM?)
  (FGETSEL TSTREAM? SET))
  then (CL:WHEN [type? TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of (SETQ X (FGETSEL TSTREAM?
    SELTEXTSTREAM)
    X))])
(OR TS (CL:UNLESS NOERROR (ERROR TSTREAM? "is not a Tedit document"))

```

(TEXTSTREAMP

[LAMBDA (TSTREAM)

; Edited 20-Mar-2024 07:55 by rmk
(* jds " 3-Apr-84 14:34")

:: Returns the TSTREAM if it is a text stream, else NIL. Use TEXTSTREAM for coercion.

```

(CL:WHEN (type? TEXTSTREAM TSTREAM)
  TSTREAM])

```

(TEDITMENU

[LAMBDA (WINDOW TITLE)

; Edited 15-Mar-2024 15:39 by rmk
; Edited 7-Dec-2023 21:06 by rmk
; Edited 20-Sep-2023 22:36 by rmk
; Edited 10-Apr-2023 10:14 by rmk

```

(CL:WHEN (AND (WINDOWP WINDOW)
  (WINDOWPROP WINDOW 'TEDITMENU)
  (fetch (TEXTWINDOW WTEXTOBJ) of WINDOW)
  (CL:IF TITLE
    (STRING.EQUAL TITLE (WINDOWPROP WINDOW 'TITLE))
    T))
  (WINDOWPROP WINDOW 'TITLE)))

```

(COERCETEXTSTREAM

[LAMBDA (TSTREAM TYPE OUTPUTSTREAM)

; Edited 17-Mar-2024 12:05 by rmk
; Edited 13-Jan-2024 20:01 by rmk
; Edited 26-Dec-2023 12:29 by rmk
; Edited 18-Dec-2023 23:13 by rmk
; Edited 21-Nov-2023 00:08 by rmk
; Edited 15-Sep-2023 00:08 by rmk
; Edited 15-Aug-2023 20:20 by rmk
; Edited 8-May-2023 13:25 by rmk
; Edited 4-May-2023 12:13 by rmk
; Edited 11-Jun-99 15:10 by rmk:
; Edited 18-Apr-93 23:42 by jds

:: Coerce the contents of the TEXOTBJ to be of the given type. This is for making a string from a textobj, e.g.
:: RMK: moved the string case up from the piece level to the stream-BIN level

```

(LET ((TEXTOBJ (TEXTOBJ TSTREAM))
  (SETQ TSTREAM (TEXTSTREAM TEXTOBJ))
  (SELECTQ TYPE
    ((STRINGP STRING)
      (\TEDIT.TEXTSETFILEPTR TSTREAM 0)

```

; This gets underneath the OPENP test of the generic
; SETFILEPTR

:: Allocstring presumably errors if greater than max stringsize.

```

[for CH# CH (NOBJECTS _ 0)
  STR _ (ALLOCSTRING (TEXTLEN TEXTOBJ))] from 1 to (TEXTLEN TEXTOBJ)
do (SETQ CH (BIN TSTREAM))
  (if (IMAGEOBJP CH)
    then (add NOBJECTS 1) ; Skip image objects
    (add CH# -1)
  else (RPLCHARCODE STR CH# CH))
finally (RETURN (OR (SUBSTRING STR 1 (IDIFFERENCE CH# (ADD1 NOBJECTS)))
  (CONCAT "")))

```

(STREAM ; It seems that all this does is to ensure that the TXTFILE is open and TSTREAM is set to the beginning

```

(CL:WHEN (STREAMP (FGETTOBJ TEXTOBJ TXTFILE))
  (CL:UNLESS (\GETSTREAM (FGETTOBJ TEXTOBJ TXTFILE))
    (OPENSTREAM (FGETTOBJ TEXTOBJ TXTFILE)
      'INPUT))
  (SETFILEPTR TSTREAM 0)
  TSTREAM)

```

(FILE ; Throw away looks if no formatting

```

[SETQ OUTPUTSTREAM (OR (AND OUTPUTSTREAM (OPENP OUTPUTSTREAM 'OUTPUT))
  (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW)
  (\TEDIT.PUT.PCTB TEXTOBJ OUTPUTSTREAM (NOT (\TEDIT.FORMATTEDSTREAMP TEXTOBJ)))
  OUTPUTSTREAM)

```

(SPLIT ; Return 2 NODIRCORE files, one with the plain text character, one with formatting info, such that concatenating them will
:: produce a proper Tedit-format file. If TSTREAM is actually a file stream and not a text stream, we could do the split by
:: COPYBYTES. But if a text stream, there could be other kinds of pieces.

```
(LET [(CHARSTREAM (OPENSTREAM '{NODIRCORE}' BOTH 'NEW))
      (FORMATSTREAM (OPENSTREAM '{NODIRCORE}' BOTH 'NEW' ((LINELENGTH T)
        (\TEDIT.PUT.PCTB TEXTOBJ CHARSTREAM FORMATSTREAM NIL T)
        (SETFILEPTR CHARSTREAM 0)
        (SETFILEPTR FORMATSTREAM 0)
        (CONS CHARSTREAM FORMATSTREAM)))
      (SEXPR (TEDIT.SEL.AS.SEXPR TSTREAM)
        NIL])
```

(TEDIT.CONCAT

```
[LAMBDA (TSTREAMS SEPARATOR)
```

```
; Edited 8-Feb-2025 20:58 by rmk
; Edited 17-Mar-2024 00:21 by rmk
; Edited 18-Jan-2024 00:03 by rmk
```

```
:: Produces a textstream that contains the concatenation of all of the TSTREAMS, separated by SEPARATOR. Any stream that is not already a
:: text stream is first converted to a plaintext stream. SEPARATOR if provided as a string or character is inserted between the files.
```

```
(CL:WHEN SEPARATOR
  (CL:UNLESS (CHARCODEP SEPARATOR)
    (SETQ SEPARATOR (OR (CHARCODE.DECODE SEPARATOR T)
      (MKSTRING SEPARATOR))))))
(LET* ((CSTREAM (OPENTEXTSTREAM))
       (CTEXTOBJ (TEXTOBJ CSTREAM))
       [TSTEXTOBJECTS (for TS inside TSTREAMS collect (OR (TEXTOBJ TS T)
        (TEXTOBJ (OPENTEXTSTREAM TS)
          FIRSTTOBJ INITIALFILEPIECES)
        (CL:WHEN TSTEXTOBJECTS
          (SETQ FIRSTTOBJ (CAR TSTEXTOBJECTS))
          ;; Take overall parameters from the first stream.
          (FSETTOBJ CTEXTOBJ DEFAULTCHARLOOKS (FGETTOBJ FIRSTTOBJ DEFAULTCHARLOOKS))
          (FSETTOBJ CTEXTOBJ DEFAULTPARALOOKS (FGETTOBJ FIRSTTOBJ DEFAULTPARALOOKS))
          (FSETTOBJ CTEXTOBJ TXTRTBL (FGETTOBJ FIRSTTOBJ TXTRTBL))
          (FSETTOBJ CTEXTOBJ TXTWTBL (FGETTOBJ FIRSTTOBJ TXTWTBL))
          (FSETTOBJ CTEXTOBJ TXTSTYLESHEET (FGETTOBJ FIRSTTOBJ TXTSTYLESHEET))
          (for TSOBJ PREVPC (LASTTOBJ _ (CAR (LAST TSTEXTOBJECTS)))
            (FIRSTPC _ (create PIECE)) in TSTEXTOBJECTS first ;; LASTTOBJ to suppress final separator
              (SETQ PREVPC FIRSTPC)
              ; Dummy
          do (CL:WHEN (FGETTOBJ TSOBJ FORMATTEDP)
            (FSETTOBJ CTEXTOBJ FORMATTEDP T))
            (for PC NEWPIECE inpieces (\TEDIT.FIRSTPIECE TSOBJ) do (SETQ NEWPIECE
              (\TEDIT.COPYPIECE PC TSOBJ CTEXTOBJ
                NIL 'COPY))
              (FSETPC PREVPC NEXTPIECE NEWPIECE)
              (FSETPC NEWPIECE PREVPIECE PREVPC)
              (SETQ PREVPC NEWPIECE))
            ;; Information for pageframe adjustments
            (push INITIALFILEPIECES (\TEDIT.FIRSTPIECE TSOBJ))
            (CL:WHEN SEPARATOR
              (CL:UNLESS (EQ TSOBJ LASTTOBJ)
                (SETQ PREVPC (\TEDIT.MAKE.STRINGPIECE PREVPC SEPARATOR))))
            finally (\TEDIT.INSERTPIECES (NEXTPIECE FIRSTPC)
              NIL CTEXTOBJ)
              (\TEDIT.CONCAT.PAGEFRAMES CTEXTOBJ TSTEXTOBJECTS (DREVERSE INITIALFILEPIECES))
              (\TEDIT.UNIQUIFY.ALL CTEXTOBJ))
          CSTREAM])
```

(TEDITSTRING

```
[LAMBDA (TEXT WINDOW DONTSPAWN PROPS)
```

```
;;; Edited 20-Feb-2025 08:50 by rmk
```

```
;;; Edited 31-Mar-2024 10:13 by rmk: If TEXT is NIL, don't coerce it to "NIL"
```

```
;;; Edited 31-Mar-2024 10:12 by rmk
```

```
;;; Edited 9-May-2023 21:55 by rmk
```

```
;;; Edited 23-May-2022 15:52 by rmk
```

```
;;; Edited 19-May-2022 22:46 by rmk: An interface function to replace calls to TEDIT when the text argument may be the string to be edited rather than
;;; the name of a file. This enables the transition that gets TEDIT aligned with the convention that strings, as well as litatoms, are file names.
```

```
(CL:WHEN TEXT
  (SETQ TEXT (MKSTRING TEXT)))
(TEDIT (LET ((TSTR (OPENTEXTSTREAM NIL NIL PROPS)))
  (TEDIT.INSERT TSTR TEXT 1 NIL T)
  (TEDIT.SETSEL TSTR 1 0 'LEFT)
  TSTR)
  WINDOW DONTSPAWN PROPS])
```

(TEDIT-SEE

[LAMBDA (FILE WINDOW FORMAT TITLE)

:: Edited 13-Sep-2023 09:04 by rmk: Old code replaced to take advantage of new standard interfaces.
:: Edited 14-Jul-2023 00:02 by rmk
:: Edited 13-Sep-2022 09:21 by rmk: Changed the default format here to the current Medley default. This shouldn't be special.
:: Edited 27-Feb-2021 20:07 by rmk:
:: Edited 1-Feb-88 19:00 by bvm:

(TEXTSTREAM (TEDIT FILE WINDOW NIL `(READONLY T LEAVETTY T FONT ,DEFAULTFONT))

(TEDIT.COPY

[LAMBDA (FROM TO)

; Edited 2-Dec-2024 09:02 by rmk
; Edited 7-Jul-2024 16:09 by rmk
; Edited 2-Jul-2024 10:40 by rmk
; Edited 18-May-2024 16:21 by rmk
; Edited 12-May-2024 20:54 by rmk
; Edited 22-Apr-2024 23:55 by rmk
; Edited 29-Apr-2024 12:54 by rmk
; Edited 15-Mar-2024 13:54 by rmk
; Edited 20-Feb-2024 17:03 by rmk
; Edited 1-Feb-2024 20:37 by rmk
; Edited 20-May-2023 18:47 by rmk
; Edited 15-May-2023 22:11 by rmk
; Edited 4-Jun-92 11:11 by jds

:: Copy the FROM-selected pieces into the destination object and position as indicated by the TO selection. FROM and TO are external selections,
:: with SELTEXTSTREAMS.

:: This results in a single history event, either :Insert or :Replace depending on bluependingdelete.

(LET ((FROMSTREAM (TEXTSTREAM FROM))
(TOSTREAM (TEXTSTREAM TO)))
(CL:UNLESS (type? SELECTION FROM)
(SETQ FROM (TEXTSEL (GETTSTR FROMSTREAM TEXTOBJ))))
(CL:UNLESS (type? SELECTION TO)
(SETQ TO (TEXTSEL (GETTSTR TOSTREAM TEXTOBJ))))
(CL:UNLESS (EQ TO FROM)
(\TEDIT.COPY FROM TO FROMSTREAM TOSTREAM]))

(TEDIT.DELETE

[LAMBDA (TSTREAM SEL LEN LEAVECARETLOOKS)

; Edited 22-Jun-2024 00:06 by rmk
; Edited 22-May-2024 09:44 by rmk
; Edited 23-May-2023 12:57 by rmk
; Edited 22-May-2023 10:54 by rmk
; Edited 10-Nov-2022 22:48 by rmk
; Edited 12-Jun-90 17:49 by mitani

:: Delete the specified characters from TSTREAM.

:: If LEAVECARETLOOKS is non-NIL, the selection will NOT be set up to do the right thing with type-in. This can save time in inner loops.

(LET ((TEXTOBJ (TEXTOBJ TSTREAM))
(if (FIXP SEL)
then (TEDIT.SETSEL TSTREAM SEL LEN NIL NIL LEAVECARETLOOKS)
(SETQ SEL (FGETTOBJ TEXTOBJ SEL))
elseif (NULL SEL)
then (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
(SELECTION! SEL)
(\TEDIT.DELETE TEXTOBJ SEL]))

(TEDIT.INSERT

[LAMBDA (TSTREAM TEXT CH#ORSEL LOOKS DONTSCROLL)

; Edited 2-Aug-2024 22:17 by rmk
; Edited 31-Jul-2024 12:13 by rmk
; Edited 23-Jul-2024 16:35 by rmk
; Edited 7-Jul-2024 12:33 by rmk
; Edited 22-Jun-2024 00:02 by rmk
; Edited 20-Jun-2024 09:08 by rmk
; Edited 22-May-2024 14:00 by rmk
; Edited 22-Dec-2023 22:05 by rmk
; Edited 12-Nov-2023 12:30 by rmk
; Edited 15-Oct-2023 14:57 by rmk
; Edited 31-May-2023 23:25 by rmk
; Edited 27-May-2023 10:47 by rmk
; Edited 9-Nov-2022 10:36 by rmk
; Edited 29-May-91 18:21 by jds

(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(if (IMAGEOBJP TEXT)
then (TEDIT.INSERT.OBJECT TEXT TSTREAM CH#ORSEL LOOKS)
else (CL:WHEN (ATOM TEXT)
(SETQ TEXT (MKSTRING TEXT)))
(CL:WHEN (AND (STRINGP TEXT)
(NEQ 0 (NCHARS TEXT))))

:: Nothing to do for an empty string

(LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))

```

    (if (FIXP CH#ORSEL)
      then (TEDIT.SETSEL TEXTOBJ CH#ORSEL 1 'LEFT)
           ; He gave us a ch# to insert before
           (SETQ CH#ORSEL (TEXTSEL TEXTOBJ))
      elseif (NOT CH#ORSEL)
      then (SETQ CH#ORSEL (TEXTSEL TEXTOBJ))
    (SELECTION! CH#ORSEL)
    (if (FGETSEL CH#ORSEL SET)
      then (\TEDIT.INSERT TEXT CH#ORSEL TSTREAM DONTSCROLL)
           ; TEXTSEL now selects the insertion, apply the looks.
           (CL:WHEN LOOKS (\TEDIT.CHANGE.CHARLOOKS TSTREAM LOOKS))
      else (TEDIT.PROMPTPRINT TEXTOBJ "Please select a place for the insertion." T))))

```

(TEDIT.TERPRI

```

[LAMBDA (TSTREAM CH#ORSEL DONTSCROLL)
  ; Edited 12-Aug-2024 20:04 by rmk
  ;; Inserts an EOL at CH#ORSEL, and then marks that as the end of a paragraph. Unlike BOUT, doesn't replace the character currently at that
  ;; position, inserts in front. \TEDIT.INSERT will also clear out any bluependingdelete, and manage the display update.
  (SETQ TSTREAM (TEXTSTREAM TSTREAM))
  (LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))
    (if (FIXP CH#ORSEL)
      then (TEDIT.SETSEL TEXTOBJ CH#ORSEL 1 'LEFT)
           ; He gave us a ch# to insert before
           (SETQ CH#ORSEL (TEXTSEL TEXTOBJ))
      elseif (NOT CH#ORSEL)
      then (SETQ CH#ORSEL (TEXTSEL TEXTOBJ))
    (SELECTION! CH#ORSEL)
    (if (FGETSEL CH#ORSEL SET)
      then (\TEDIT.INSERT [CONSTANT (CONCATCODES (CONS (CHARCODE EOL)
                                                         CH#ORSEL TSTREAM DONTSCROLL))
                           (DEL.PROCESS TEDIT)]
           ; He gave us a ch# to insert before
           (CL:WHEN LOOKS (\TEDIT.CHANGE.CHARLOOKS TSTREAM LOOKS))
      else (TEDIT.PROMPTPRINT TEXTOBJ "Please select a place for the insertion." T]))

```

(TEDIT.KILL

```

[LAMBDA (TSTREAM)
  ; Edited 25-Jun-2024 11:59 by rmk
  ; Edited 12-May-2024 11:55 by rmk
  ; Edited 20-Sep-2023 17:55 by rmk
  ; Edited 12-Jun-90 17:49 by mitani

  ;; Force the edit session supported by TSTREAM to terminate
  (LET ((TEXTOBJ (TEXTOBJ TSTREAM))
        TEDW TEDPROC)
    (FSETTOBJ TEXTOBJ EDITFINISHEDFLG T)
    (CL:WHEN (AND (SETQ TEDW (\TEDIT.PRIMARYPANE TEXTOBJ))
                  [PROCESSP (SETQ TEDPROC (WINDOWPROP TEDW 'PROCESS])
                            (NEQ TEDPROC (THIS.PROCESS))])
              (DEL.PROCESS TEDPROC)
              (TEDIT.DEACTIVATE.WINDOW TEDW))))

```

(TEDIT.QUIT

```

[LAMBDA (TSTREAM VALUE)
  ; Edited 12-Feb-2025 16:26 by rmk
  ; Edited 9-Feb-2025 21:22 by rmk
  ; Edited 7-Feb-2025 23:45 by rmk
  ; Edited 29-Jun-2024 09:12 by rmk
  ; Edited 25-Jun-2024 11:59 by rmk
  ; Edited 20-Sep-2023 17:55 by rmk
  ; Edited 10-Apr-2023 10:19 by rmk
  ; Edited 12-Jun-90 17:49 by mitani

  ;; Force the edit session supported by TSTREAM to terminate, and to return VALUE
  (LET* ((TEXTOBJ (TEXTOBJ TSTREAM))
         (PRIMPANE (FGETTOBJ TEXTOBJ PRIMARYPANE)))
    ;; Make sure the process has the TTY, then tell the command loop to finish.
    (CL:WHEN (AND PRIMPANE (WINDOWPROP PRIMPANE 'PROCESS))
              (TTY.PROCESS (WINDOWPROP PRIMPANE 'PROCESS)))
    (FSETTOBJ TEXTOBJ EDITFINISHEDFLG (OR VALUE T)))

```

(TEDIT.MOVE

```

[LAMBDA (FROM TO)
  ; Edited 2-Dec-2024 09:02 by rmk
  ; Edited 2-Jul-2024 14:11 by rmk

  ;; Public entry for moving FROM-selected text in its document to the TO-selected position in its document. FROM and TO may eventually be
  ;; user-level selections that need to be converted here to internal SELECTION data structures.
  (LET ((FROMSTREAM (TEXTSTREAM FROM))
        (TOSTREAM (TEXTSTREAM TO)))
    (CL:UNLESS (type? SELECTION FROM)
      (SETQ FROM (TEXTSEL (GETTSTR FROMSTREAM TEXTOBJ))))
    (CL:UNLESS (type? SELECTION TO)
      (SETQ TO (TEXTSEL (GETTSTR TOSTREAM TEXTOBJ))))
    (CL:UNLESS (EQ TO FROM)
      (\TEDIT.MOVE FROM TO FROMSTREAM TOSTREAM)))

```

(TEDIT.STRINGWIDTH

```
[LAMBDA (STR FONT TERMSA)
(COND
  (TERMSA
    (for CH instring STR sum (TEDIT.CHARWIDTH CH FONT TERMSA)))
  (T
    (for CH instring STR sum (SELCHARQ CH
      (TAB 36)
      (CHARWIDTH CH FONT))
      (* jds "19-AUG-83 14:40")
      (* We have a terminal table to take account of.
      Do so.)
      (* Just use the native character widths)
```

(TEDIT.CHARWIDTH

```
[LAMBDA (CH FONT TERMSA)
; Edited 21-Oct-2024 00:33 by rmk
(* jds "22-OCT-83 19:32")
(* Returns the width of CH in FONT printed according to any special printing instructions in CHARTABLE TERMSA)
```

```
(COND
  (TERMSA
    (SELECTC (fetch CCECHO of (\SYNCODE TERMSA CH))
      (INDICATE.CCE (IPLUS (COND
        ((IGREATERP CH 127) (* META character)
          (SETQ CH (LOGAND CH 127))
          (CHARWIDTH (CHARCODE %#)
            FONT))
        (T 0))
        (COND
          ((ILESSP CH 32) (* CONTROL character)
            (SETQ CH (LOGOR CH 64))
            (CHARWIDTH (CHARCODE ^)
              FONT))
          (T 0))
        (CHARWIDTH CH FONT)))
      (SIMULATE.CCE (SELCHARQ CH
        ((EOL CR LF)
          (IMAX 6 (CHARWIDTH CH FONT)))
        (ESCAPE (CHARWIDTH (CHARCODE $)
          FONT))
        (BELL 0)
        (TAB 36)
        (CHARWIDTH CH FONT)))
      (REAL.CCE (CHARWIDTH CH FONT))
      (IGNORE.CCE 0)
      (\TEDIT.THELP)))
    (* There IS a TERMTABLE to account for)
    (* The usual case is to treat every character as a graphic.)
  (T
    (SELCHARQ CH
      (CR (IMAX 6 (CHARWIDTH CH FONT)))
      (TAB 36)
      (CHARWIDTH CH FONT))
    )
```

(DEFINEQ

(TEXTOBJ

```
[LAMBDA (TEXTOBJ? NOERROR)
; Edited 20-Mar-2024 07:36 by rmk
; Edited 12-Jun-90 17:50 by mitani
(if (type? TEXTOBJ TEXTOBJ?)
  then TEXTOBJ?
  else (LET ((TSTRM (TEXTSTREAM TEXTOBJ? NOERROR)))
    (CL:WHEN TSTRM
      (fetch (TEXTSTREAM TEXTOBJ) of TSTRM))))
```

(COERCETEXTOBJ

```
[LAMBDA (TSTREAM TYPE OUTPUTSTREAM)
; Edited 20-Mar-2024 08:02 by rmk
; Edited 17-Mar-2024 12:05 by rmk
; Edited 13-Jan-2024 20:01 by rmk
; Edited 26-Dec-2023 12:29 by rmk
; Edited 18-Dec-2023 23:13 by rmk
; Edited 21-Nov-2023 00:08 by rmk
; Edited 15-Sep-2023 00:08 by rmk
; Edited 15-Aug-2023 20:20 by rmk
; Edited 8-May-2023 13:25 by rmk
; Edited 4-May-2023 12:13 by rmk
; Edited 11-Jun-99 15:10 by rmk:
; Edited 18-Apr-93 23:42 by jds
```

:: Documented entry, to be deprecated in favor of COERCEDTEXTSTREAM.

```
(COERCETEXTSTREAM TSTREAM TYPE OUTPUTSTREAM])
```

)

(DECLARE%: EVAL@COMPILE

```
(PUTPROPS TEVAL MACRO [(FORM WINDOW TIT DONTDEFER)
  (LET* [(TSTREAM (TEXTSTREAM (TEDIT NIL (OR WINDOW 'TeditEval)
; Capture FORM's T output in Tedit
```



```

NIL
\ (LEAVETTY T HISTORY OFF PARABREAKCHARS NIL APPEND
  QUIET TITLE , (OR TIT "Tedit EVAL")
  , (PACK* "F" "ORM")
  'FORM FONT DEFAULTFONT]

```

```

(RESETLST
  (RESETSAVE (TTYDISPLAYSTREAM TSTREAM))
  [RESETSAVE (DSPFONT DEFAULTFONT T)
    ' (PROGN (DSPFONT OLDVALUE T)
      (BKSYSBUF " " ) ; Suppress pagehold
      (CL:UNLESS DONTDEFER (TEDIT.DEFER.UPDATES TSTREAM))
      (CL:UNLESS TIT
        (PRINTDEF 'FORM NIL T NIL NIL T)
        (TERPRI T))
      (PROG1 FORM (TERPRI T)))]])

```

```

)
(DEFINEQ

```

(TDRIBBLE
[LAMBDA NIL

```

; Edited 27-Nov-2024 23:20 by rmk
; Edited 17-Nov-2024 14:10 by rmk
; Edited 15-Nov-2024 21:13 by rmk
; Edited 22-Oct-2024 21:23 by rmk

```

```

(LET [(TSTREAM (OPENTEXTSTREAM NIL NIL NIL NIL '(HISTORY OFF TITLE "Tedit dribble" FONT DEFAULTFONT)
  [WHENCLOSE TSTREAM 'BEFORE (FUNCTION (LAMBDA (TSTREAM)
    (TEDIT TSTREAM 'TeditDribble NIL
      ' (LEAVETTY T APPEND QUIET PARABREAKCHARS NIL HISTORY OFF)
    )
    (TEDIT.SETSEL TSTREAM 1 0)
  )
  (DRIBBLE TSTREAM)])

```

:: Object-oriented editing

```

(DEFINEQ

```

(TEDIT.INSERT.OBJECT

[LAMBDA (OBJECT TSTREAM CH# LOOKS)

```

; Edited 25-Feb-2025 11:18 by rmk
; Edited 2-Feb-2025 11:37 by rmk
; Edited 26-Dec-2024 10:13 by rmk
; Edited 21-Oct-2024 00:26 by rmk
; Edited 2-Aug-2024 08:46 by rmk
; Edited 30-Jul-2024 22:19 by rmk
; Edited 23-Jul-2024 22:20 by rmk
; Edited 7-Jul-2024 12:32 by rmk
; Edited 22-May-2024 13:56 by rmk
; Edited 18-May-2024 16:20 by rmk
; Edited 12-May-2024 20:53 by rmk
; Edited 3-Mar-2024 13:01 by rmk
; Edited 16-Mar-2024 00:08 by rmk
; Edited 9-Feb-2024 10:52 by rmk
; Edited 28-Jan-2024 23:29 by rmk
; Edited 11-Dec-2023 08:21 by rmk
; Edited 12-Nov-2023 12:16 by rmk
; Edited 19-May-2023 00:18 by rmk
; Edited 21-Apr-93 00:52 by jds

```

:: Inserts the Image-object OBJECT into text STREAM in front of character CH#.

```

(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(PROG ((TEXTOBJ (TEXTOBJ TSTREAM))
  SEL OBJPC OBJSELPIECES SUBSTREAM)

```

:: We construct and copy a trivial SELPIECES so that we can share the basic insertion code.

```

(CL:UNLESS CH#
  (SETQ CH# (TEXTSEL TEXTOBJ)))
(CL:WHEN (type? SELECTION CH#)
  (SETQ CH# (TEDIT.GETPOINT TSTREAM CH#)))
(CL:WHEN (\TEDIT.READONLY TSTREAM NIL CH#)
  (RETURN))
(SETQ OBJPC (create PIECE
  PTYPE _ OBJECT.PTYPE
  PCONTENTS _ OBJECT
  PLEN _ 1
  PLOOKS _ (FGETTOBJ TEXTOBJ CARETLOOKS)))
; The new piece we're inserting
(CL:WHEN (SETQ SUBSTREAM (IMAGEOBJPROP OBJECT 'SUBSTREAM))
  ; If this is computed text in bulk, fix the length.

```

```

  (\TEDIT.THELP "SUBSTREAM NOT IMPLEMENTED")
  (FSETPC OBJPC PTYPE SUBSTREAM.PTYPE)
  (FSETPC OBJPC PLEN (TEXTLEN (fetch (TEXTSTREAM TEXTOBJ) of SUBSTREAM))))
(SETQ OBJSELPIECES
  (\TEDIT.SELPIECES.COPY (create SELPIECES
    SPLLEN _ 1
    SPFIRST _ OBJPC
    SPLAST _ OBJPC

```

```

SPFIRSTCHAR _ CH#
SPLASTCHAR _ CH#
      'INSERT TEXTOBJ))
(CL:UNLESS OBJSELPIECES
  (RETURN))
; Copy may not be allowed

```

;;
 ;; OBJSELPIECES contains (a copy of) the object piece, and the object approved of insertion.

```

(SETQ SEL (TEXTSEL TEXTOBJ))
(\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
(CL:WHEN (type? SELECTION CH#)
  (SETQ CH# (GETSEL CH# CH#)))
(CL:WHEN (FGETTOBJ TEXTOBJ BLUEPENDINGDELETE)
  (\TEDIT.DELETE TEXTOBJ SEL))
(CL:WHEN CH#
  (\TEDIT.UPDATE.SEL SEL (IMIN CH# (ADD1 (TEXTLEN TEXTOBJ)))
    0
    'LEFT)
  (\TEDIT.FIXSEL SEL TEXTOBJ))
(\TEDIT.INSERT.SELPIECES OBJSELPIECES TEXTOBJ SEL)
(CL:WHEN LOOKS (\TEDIT.CHANGE.CHARLOOKS TSTREAM LOOKS SEL))
(\TEDIT.SCROLL.CARET TSTREAM)
(\TEDIT.SHOWSEL SEL T TEXTOBJ])

```

(TEDIT.EDIT.OBJECT

```
[LAMBDA (TSTREAM OBJ)
```

```

; Edited 3-Oct-2024 22:08 by rmk
; Edited 10-May-2024 22:42 by rmk
; Edited 7-May-2024 08:18 by rmk
; Edited 29-Apr-2024 12:41 by rmk
; Edited 15-Mar-2024 14:23 by rmk
; Edited 2-Dec-2023 09:57 by rmk
; Edited 19-May-2023 21:35 by rmk
; Edited 27-Apr-2023 00:14 by rmk
; Edited 21-Oct-2022 18:37 by rmk
; Edited 29-May-91 18:23 by jds

```

```

(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(LET ((TEXTOBJ (TEXTOBJ! (FGETTSTR TSTREAM TEXTOBJ)))
  SEL CH# EDITFN)
  [COND
    [(AND OBJ (IMAGEOBJP OBJ))
      (SETQ CH# (TEDIT.FIND.OBJECT TEXTOBJ OBJ))
      (COND
        (CH# (SETQ SEL (\TEDIT.COPYSEL (FGETTOBJ TEXTOBJ SEL)))
          (\TEDIT.UPDATE.SEL SEL CH# 1)
          (SETSEL SEL SELOBJ OBJ)
          (\TEDIT.FIXSEL SEL TEXTOBJ))
        (T (TEDIT.PROMPTPRINT TEXTOBJ "Can't find specified object." T)
          (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
          (SETQ OBJ (GETSEL SEL SELOBJ))
          (COND
            (CH#
              (SETQ EDITFN (IMAGEOBJPROP OBJ 'EDITFN))
              (CL:UNLESS (AND EDITFN (APPLY* EDITFN OBJ))
                (TEDIT.OBJECT.CHANGED TSTREAM OBJ))
              (T (TEDIT.PROMPTPRINT TEXTOBJ "Please select an editable object" T T]))
            )
          )
        )
      ]

```

```

; OK There's an object selected. Edit it.
; If the editfn makes a change, update the screen.

```

(TEDIT.FIND.OBJECT

```
[LAMBDA (TSTREAM OBJ START END)
```

```

; Edited 20-Oct-2024 12:07 by rmk
; Edited 10-May-2024 21:58 by rmk
; Edited 16-Mar-2024 10:03 by rmk
; Edited 6-Nov-2022 11:12 by rmk
; Edited 3-May-93 12:52 by jds

```

;; Return the character number of OBJ in TSTREAM, if it occurs between START and END. We know that an object occupies its own singleton piece, so we don't need to worry about starting or ending in the middle of a piece. We also don't need to test PTYPE, just look at PCONTENTS.

```

(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(CL:WHEN (IMAGEOBJP OBJ)
  [LET ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ)))
    (CL:UNLESS END
      (SETQ END (FGETTOBJ TEXTOBJ TEXTLEN)))
    (CL:UNLESS START
      (SETQ START (TEDIT.GETPOINT TSTREAM)))
    (CL:WHEN (AND (ILEQ START END)
      (SETQ START (\TEDIT.CHTOPC START TEXTOBJ)))
      (SETQ END (\TEDIT.CHTOPC END TEXTOBJ))
      (for PC inpieces START when (EQ OBJ (PCONTENTS PC)) do (RETURN (\TEDIT.PCTOCH PC TEXTOBJ))
        repeatuntil (EQ PC END)))))]

```

(TEDIT.FIND.OBJECT.BACKWARD

```
[LAMBDA (TSTREAM OBJ START END AGAIN)
```

```

; Edited 10-May-2024 22:06 by rmk
; Edited 16-Mar-2024 10:03 by rmk
; Edited 6-Nov-2022 11:12 by rmk
; Edited 3-May-93 12:52 by jds

```

```
;; Return the character number of OBJ in TSTREAM, if it occurs between START and END and is the occurrence closest to END. START defaults
;; to 1, END defaults to current caret position (or one before, if AGAIN).

;; If we were sure that a given object can appear only once in a document, we could just run the TEDIT.FIND.OBJECT with different defaults for
;; START and END, but...

;; We know that an object occupies its own singleton piece, so we don't need to worry about starting or ending in the middle of a piece. We also
;; don't need to test PTYPE, just look at PCONTENTS.
```

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(CL:WHEN (IMAGEOBJ OBJ)
  [LET [(TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)
    (SETQ START (IMAX 1 (OR START 1)))
    (SETQ END (IMIN (OR END (SUB1 (TEDIT.GETPOINT TSTREAM)))
      (FGETTOBJ TEXTOBJ 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)
      (SETQ START (\TEDIT.CHTOPC START TEXTOBJ))
      (SETQ END (\TEDIT.CHTOPC END TEXTOBJ))
      (for PC backpieces END when (EQ OBJ (PCONTENTS PC)) do (RETURN (\TEDIT.PCTOCH PC TEXTOBJ))
        repeatuntil (EQ PC START)))]])]
```

(TEDIT.OBJECT.CHANGED

```
[LAMBDA (TSTREAM OBJECT PIECE/CH#/SEL)
  ; Edited 26-Nov-2024 03:52 by rmk
  ; Edited 20-Oct-2024 12:08 by rmk
  ; Edited 19-Oct-2024 10:03 by rmk
  ; Edited 3-Oct-2024 22:58 by rmk
  ; Edited 16-Aug-2024 10:11 by rmk
  ; Edited 18-May-2024 17:13 by rmk
  ; Edited 10-May-2024 22:42 by rmk
  ; Edited 7-May-2024 08:18 by rmk
  ; Edited 17-Mar-2024 00:25 by rmk
  ; Edited 21-Oct-2023 08:59 by rmk
  ; Edited 18-Apr-2023 23:57 by rmk
  ; Edited 10-Apr-2023 00:02 by rmk
  ; Edited 9-Sep-2022 09:32 by rmk
  ; Edited 6-Aug-2022 09:37 by rmk
  ; Edited 12-Jun-90 17:51 by mitani
```

```
;; Notifies TEdit that an object has changed, and the display may need to be updated. Caller may provide the CH# or PIECE containing the object,
;; otherwise we scan.
```

```
(LET ((TEXTOBJ (TEXTOBJ TSTREAM))
  CH#)
  (SETQ CH# (if (FIXP PIECE/CH#/SEL)
    elseif (type? PIECE PIECE/CH#/SEL)
      then (CL:WHEN (EQ OBJECT (POBJ PIECE/CH#/SEL))
        (\TEDIT.PCTOCH PIECE/CH#/SEL TEXTOBJ))
    elseif (AND (type? SELECTION PIECE/CH#/SEL)
      (EQ OBJECT (FGETSEL PIECE/CH#/SEL SELOBJ)))
      then (FGETSEL PIECE/CH#/SEL CH#)
    else (TEDIT.FIND.OBJECT TSTREAM OBJECT 1)))
  (if CH#
    then
      ; Change affected lines
      (\TEDIT.UPDATE.LINES TEXTOBJ 'CHANGED CH# 1)
      ; And mark the document dirty.
      (\TEDIT.SHOWSEL NIL T TEXTOBJ)
      (FSETTOBJ TEXTOBJ \DIRTY T)
    else (TEDIT.PROMPTPRINT TSTREAM "Changed object not found in document" T])
```

(TEDIT.MAP.OBJECTS

```
[LAMBDA (TSTREAM FN FNARG COLLECT?)
  ; Edited 25-Feb-2025 15:06 by rmk
  ; Edited 23-Apr-2024 09:15 by rmk
  ; Edited 16-Mar-2024 10:03 by rmk
  ; Edited 4-Mar-2024 16:12 by rmk
  ; Edited 6-Nov-2022 12:15 by rmk
```

```
;; Apply FN to each of the imageobjects in TSTREAM. If COLLECT? value is the list of (CH# OBJ FNVAL) pairs that satisfy the predicate
;; FN is a function of 3 args ( CH#-of-OBJ OBJ FNARG). FN defaults to TRUE
```

```
(CL:UNLESS FN
  (SETQ FN (FUNCTION TRUE)))
(for CH# OBJ FNVAL from 1 by (PLEN PC) as PC inpieces (\TEDIT.FIRSTPIECE (TEXTOBJ TSTREAM))
  when (AND (EQ OBJECT.PTYPE (PTYPE PC))
    (type? IMAGEOBJ (SETQ OBJ (PCONTENTS PC)))
    (SETQ FNVAL (APPLY* FN CH# OBJ FNARG)))
  do (SELECTQ COLLECT?
    (NIL)
    (OBJECT (PUSH $$VAL OBJ))
    (CH# (PUSH $$VAL CH#))
    (VALUE (PUSH $$VAL CH#)
      FNVAL)
    (FIRST (RETURN (LIST CH# OBJ FNVAL)))
    (PUSH $$VAL (LIST CH# OBJ FNVAL)))
  (CL:WHEN (EQ FNVAL 'STOP)
    (GO $$OUT))
```

finally (RETURN (DREVERSE \$\$VAL))

(\TEDIT.FIRST.OBJPIECE

[LAMBDA (TEXTOBJ) ; Edited 24-Jul-2024 08:47 by rmk
(find PC in (\TEDIT.FIRSTPIECE TEXTOBJ) suchthat (EQ OBJECT.PTYPE (PTYPE PC))

(\TEDIT.NEXT.OBJPIECE

[LAMBDA (TEXTOBJ PC) ; Edited 24-Jul-2024 08:47 by rmk
(find old PC inpieces (NEXTPIECE PC) suchthat (EQ OBJECT.PTYPE (PTYPE PC))

)

(FILESLOAD IMAGEOBJ)

(DEFINEQ

(\TEDIT.CONCAT.PAGEFRAMES

[LAMBDA (CTEXTOBJ TSTEXTOBJECTS INITIALFILEPIECES) ; Edited 19-Feb-2025 13:30 by rmk
; Edited 8-Feb-2025 22:27 by rmk
; Edited 17-Mar-2024 13:20 by rmk
; Edited 16-Mar-2024 10:03 by rmk
; Edited 18-Jan-2024 22:16 by rmk

:: The individual files may have their own heading paragraphs specified in their pieces and in their pageframes. Since the heading types are global
:: for the file, we have to make sure the any conflicting heading-type names are made distinct within the combined toplevel pageframe, and that any
:: new names are propagated into the PARALOOKS of the pieces within each file.

:: Scan all the first/left/right heading frames, grouping all of the heading types by their regions.

(LET (FIRSTREGIONS LEFTREGIONS RIGHTREGIONS FIRSTNEW LEFTNEW RIGHTNEW CPAGEFRAME)

:: Index first, even, odd types by region. Keep the lists separate for the final step of building the concat pageframes.

:: If the same region is appears in both left and right headings, presumably the type-names will be different--and we maintain that difference
:: as we canonicalize.

(for TSOBJ FRAMES in TSTEXTOBJECTS do (SETQ FRAMES (GETTOBJ TSOBJ TXTPAGEFRAMES))
(SETQ FIRSTREGIONS (\TEDIT.GET.PAGE.HEADINGS (CAR FRAMES)
FIRSTREGIONS))
(SETQ LEFTREGIONS (\TEDIT.GET.PAGE.HEADINGS (CADR FRAMES)
LEFTREGIONS))
(SETQ RIGHTREGIONS (\TEDIT.GET.PAGE.HEADINGS (CADDR FRAMES)
RIGHTREGIONS)))

:: Invert these to map all encountered types with a given region into a canonical type with that region.

[for R in FIRSTREGIONS as I from 1 do (for TYPE in (CDR R) do (push FIRSTNEW (LIST TYPE
(PACK* 'HeadingF I)
(CAR R)
[for R in LEFTREGIONS as I from 1 do (for TYPE in (CDR R) do (push LEFTNEW (LIST TYPE (PACK* 'HeadingL I)
(CAR R)
[for R in RIGHTREGIONS as I from 1 do (for TYPE in (CDR R) do (push RIGHTNEW (LIST TYPE
(PACK* 'HeadingR I)
(CAR R)

:: Replace the type in each heading piece to its canonical type. Presumably the input typenames were sorted by first/left/right, so a given
:: name only appears in one of the list. So we can append.

[for PC PPARALOOKS (ALLNEW _ (APPEND FIRSTNEW LEFTNEW RIGHTNEW))
inpieces
(\TEDIT.FIRSTPIECE CTEXTOBJ) everytime (SETQ PPARALOOKS (PPARALOOKS PC))
when (EQ 'PAGEHEADING (GETPLOOKS PPARALOOKS FMTPARATYPE))
do (FSETPC PC PPARALOOKS (create PARALOOKS using PPARALOOKS FMTPARASUBTYPE _
(CADR (ASSOC (FGETPLOOKS PPARALOOKS FMTPARASUBTYPE)
ALLNEW)

:: Finally, build the pageframes for the new types and their regions. We take the page frame of the first TSOBJ as the base pattern

[SETQ CPAGEFRAME (\TEDIT.PARSE.PAGEFRAMES (\TEDIT.UNPARSE.PAGEFRAMES (FGETTOBJ (CAR TSTEXTOBJECTS)
TXTPAGEFRAMES)
(\TEDIT.CONCAT.INSTALL.HEADINGS (CAR CPAGEFRAME)
FIRSTNEW)
(\TEDIT.CONCAT.INSTALL.HEADINGS (CADR CPAGEFRAME)
LEFTNEW)
(\TEDIT.CONCAT.INSTALL.HEADINGS (CADDR CPAGEFRAME)
RIGHTNEW)
(FSETTOBJ CTEXTOBJ TXTPAGEFRAMES CPAGEFRAME)]

(\TEDIT.GET.PAGE.HEADINGS

[LAMBDA (PAGEREGION HEADLIST) ; Edited 18-Jan-2024 21:36 by rmk

:: Produces an ALIST that maps each different heading region to a list of heading types that have that region. All of those heading types can be
:: reduced to a single type.

(CL:WHEN (EQ 'PAGE (fetch (PAGEREGION REGIONFILLMETHOD) of PAGEREGION))
[for PH in (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION)
when [AND (EQ 'HEADING (fetch (PAGEREGION REGIONFILLMETHOD) of PH))
(EQ 'HEADINGTYPE (CAR (fetch (PAGEREGION REGIONLOCALINFO) of PH))
do (pushnew [CDR (OR (SASSOC (fetch (PAGEREGION REGIONSPEC) of PH)
HEADLIST)
(CAR (PUSH HEADLIST (CONS (fetch (PAGEREGION REGIONSPEC) of PH)

(CADR (fetch (PAGEREGION REGIONLOCALINFO) of PH] HEADLIST]))

(\TEDIT.CONCAT.INSTALL.HEADINGS

[LAMBDA (PAGEREGION NEWTYPES) ; Edited 18-Jan-2024 22:02 by rmk

;; Smash headings representing NEWTYPES into PAGEREGION, removing any headings previously there.

(CL:WHEN (EQ 'PAGE (fetch (PAGEREGION REGIONFILLMETHOD) of PAGEREGION))

;; Replace all the old headings with the new ones, keeping all other subboxes

;; NEWTYPES is the list that maps old types to new types. We first reduce it to just a new-type region list.

[change (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION) (NCONC (for SPF in DATUM unless (EQ 'HEADING (fetch (PAGEREGION REGIONFILLMETHOD) of SPF)) collect SPF) (for R in (for NT in NEWTYPES unless (ASSOC (CADR NT) \$\$VAL) collect (CDR NT)) collect (create PAGEREGION REGIONFILLMETHOD _ 'HEADING REGIONLOCALINFO _ (LIST 'HEADINGTYPE (CAR R)) REGIONSPEC _ (CADR R]))]

)

(DEFINEQ

(\TEDIT.MOVE.MSG

[LAMBDA (FROMOBJ TOOBJ COPYFLG) ; Edited 7-Jul-2024 16:36 by rmk ; Edited 22-May-2024 14:06 by rmk ; Edited 3-May-2024 00:04 by rmk ; Edited 29-Apr-2024 12:52 by rmk ; Edited 1-Feb-2024 23:33 by rmk ; Edited 22-May-2023 09:35 by rmk ; Edited 20-May-2023 18:53 by rmk ; Edited 15-May-2023 22:11 by rmk ; Edited 4-Jun-92 11:11 by jds

;; Check whether it is possible to insert the FROM-selected pieces into the TO-selection. Value is NON-NIL if the operation can't be performed.

(LET ((TYPE (CL:IF COPYFLG "copy" "move"))) (if (AND FROMOBJ TOOBJ) then (if (EQ FROMOBJ TOOBJ) then (\TEDIT.READONLY TOOBJ NIL (FGETSEL (FGETTOBJ TOOBJ SEL) CH#)) elseif (\TEDIT.READONLY TOOBJ "Destination") else (AND (NOT COPYFLG) (\TEDIT.READONLY FROMOBJ "Source"))) else (if FROMOBJ then (TEDIT.PROMPTPRINT TOOBJ (CONCAT "Please select a destination for the " TYPE) T T) else (TEDIT.PROMPTPRINT FROMOBJ (CONCAT "Please select a source for the " TYPE) T T) T]))

(\TEDIT.READONLY

[LAMBDA (TEXTOBJ TYPE CHNO) ; Edited 4-Jul-2024 13:40 by rmk ; Edited 25-May-2024 10:01 by rmk ; Edited 22-May-2024 13:00 by rmk ; Edited 1-Feb-2024 17:33 by rmk ; Edited 13-Nov-2023 11:26 by rmk

(SETQ TEXTOBJ (TEXTOBJ TEXTOBJ)) (if (FGETTOBJ TEXTOBJ TXTREADONLY) then (CL:UNLESS (FGETTOBJ TEXTOBJ TXTREADONLYQUIET) (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (OR TYPE "Text") " is read only--aborted") T T)) 'READONLY elseif [AND (FGETTOBJ TEXTOBJ TXTAPPENDONLY) (OR (NULL CHNO) (ILEQ CHNO (FGETTOBJ TEXTOBJ TEXTLEN))] then (CL:UNLESS (FGETTOBJ TEXTOBJ TXTREADONLYQUIET) (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (OR TYPE "Text") " is append only--aborted") T T)) 'APPENDONLY])

)

(DEFINEQ

(\TEDIT.NCHARS

[LAMBDA (TSTREAM) ; Edited 1-Feb-2024 09:11 by rmk ; Edited 7-Nov-2023 09:42 by rmk

(CL:IF (type? SELECTION TSTREAM)

```
(FGETSEL TSTREAM DCH)
(FGETTOBJ (TEXTOBJ TSTREAM)
 TEXTLEN))
```

(TEDIT.RPLCHARCODE

```
[LAMBDA (TSTREAM N NEWCHARCODE NEWCHARLOOKS DONTDISPLAY)
; Edited 9-Feb-2025 12:21 by rmk
; Edited 7-Feb-2025 08:02 by rmk
; Edited 23-Sep-2024 00:36 by rmk
; Edited 27-Aug-2024 14:49 by rmk
; Edited 31-Jul-2024 12:08 by rmk
; Edited 25-Jun-2024 11:59 by rmk
; Edited 22-May-2024 14:10 by rmk
; Edited 10-May-2024 13:23 by rmk
; Edited 8-May-2024 23:09 by rmk
; Edited 5-May-2024 20:33 by rmk
; Edited 25-Apr-2024 00:13 by rmk
; Edited 17-Mar-2024 00:24 by rmk
; Edited 29-Dec-2023 11:50 by rmk
; Edited 7-Dec-2023 16:01 by rmk
; Edited 1-Dec-2023 21:52 by rmk
; Edited 9-Nov-2023 15:53 by rmk
; Edited 4-Nov-2023 15:23 by rmk
```

;; Replaces the Nth charcode (or object) in TSTREAM with NEWCHARCODE (or object) with NEWCHARLOOKS. This is accomplished by
;; isolating the target character into a length 1 piece, then converting that into a string (or object) piece containing NEWCHAR.

;; If DONTDISPLAY, this doesn't update the display.

;; NOTE: this may introduce new pieces, so must be used carefully with other piece-based or BIN-based iterations.

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(PROG ((TEXTOBJ (TEXTOBJ! (GETTSTR TSTREAM TEXTOBJ)))
 PC OFFSET START-OF-PIECE OLDCHAR PARALAST)
(DECLARE (SPECVARS START-OF-PIECE))
(CL:WHEN (ILESSP N 0)
 (add N (ADD1 (FGETTOBJ TEXTOBJ TEXTLEN))))
(CL:WHEN (\TEDIT.READONLY TEXTOBJ)
 (RETURN NIL))
(CL:UNLESS (AND (IGEQ N 1)
 (ILEQ N (FGETTOBJ TEXTOBJ TEXTLEN)))
 (\ILLEGAL.ARG N))
(CL:UNLESS (OR (CHARCODEP NEWCHARCODE)
 (IMAGEOBJP NEWCHARCODE))
 (\ILLEGAL.ARG NEWCHARCODE))
(CL:WHEN [AND NEWCHARLOOKS (NOT (OR (FONTP NEWCHARLOOKS)
 (type? CHARLOOKS NEWCHARLOOKS])
 (\ILLEGAL.ARG NEWCHARLOOKS))
(replace (STREAM BINABLE) of TSTREAM with NIL)
(SETQ PC (\TEDIT.CHTOPC N TEXTOBJ T))
(SETQ OFFSET (ADD1 (IDIFFERENCE N START-OF-PIECE))) ; Change is at OFFSET 1
(SETQ PARALAST (MEMB NEWCHARCODE (FGETTOBJ TEXTOBJ PARABREAKCHARS)))
[if (AND (SMALLP NEWCHARCODE)
 (MEMB (PTYPE PC)
 STRING.PTYPES)
 (OR (NULL NEWCHARLOOKS)
 (EQ NEWCHARLOOKS (PLOOKS PC)))
 (NEQ PC (FGETTOBJ TEXTOBJ SUFFIXPIECE))
 (NOT PARALAST))
 then ;; Fast case: Smash a new character code into an existing string piece with same looks.
 (SETQ OLDCHAR (NTHCHARCODE (PCONTENTS PC)
 OFFSET))
 (RPLCHARCODE (PCONTENTS PC)
 OFFSET NEWCHARCODE)
 (CL:WHEN (AND (EQ THINSTRING.PTYPE (PTYPE PC))
 (IGREATERP NEWCHARCODE 255))
 (FSETPC PC PTYPE FATSTRING.PTYPE)
 (FSETPC PC PBINABLE NIL)
 (FSETPC PC PBYTESPERCHAR 2)
 (FSETPC PC PBYTELEN (UNFOLD (PLEN PC)
 2))))
 elseif [AND (IMAGEOBJP NEWCHARCODE)
 (EQ OBJECT.PTYPE (PTYPE PC))
 (OR (NULL NEWCHARLOOKS)
 (EQ NEWCHARLOOKS (PLOOKS PC))
 then (SETQ OLDCHAR (POBJ PC)) ; We know PLEN is 1
 (FSETPC PC PCONTENTS NEWCHARCODE)
 else ;; The PC that contained character N becomes the suffix of characters after N,
 (CL:UNLESS (IEQP OFFSET (PLEN PC)) ; No suffix for the last character
 ;; Chop off the suffix (essentially (\TEDIT.ALIGNEDPIECE CHNO ..) but we already have the piece
 (\TEDIT.SPLITPIECE PC OFFSET TEXTOBJ)
 (SETQ PC (PREVPIECE PC))) ; Original PC holds the suffix, new PC ends with change position.
 (CL:UNLESS (EQ OFFSET 1)
 (SETQ PC (\TEDIT.SPLITPIECE PC (SUB1 OFFSET)
 TEXTOBJ))) ; Chop off the prefix. PC is now the singleton target
 ;; N is now isolated into a one-character new piece which we smash.
```

```
(SETQ OLDCHAR (\TEDIT.PIECE.NTHCHARCODE TEXTOBJ PC 1))
(if (IMAGEOBJP NEWCHARCODE)
  then (FSETPC PC PBINABLE NIL)
        (FSETPC PC PCONTENTS NEWCHARCODE)
        (FSETPC PC PTYPE OBJECT.PTYPE)
        (FSETPC PC PBYTESPERCHAR NIL) ; Doesn't make sense for objects
        (FSETPC PC PBYTELEN NIL)
  else (FSETPC PC PCONTENTS (MKSTRING (CHARACTER NEWCHARCODE))) ; Use the extend-string in INSERTCH for repeated calls?
        (if (IGREATERP NEWCHARCODE 255)
          then (FSETPC PC PTYPE FATSTRING.PTYPE)
                (FSETPC PC PBINABLE NIL)
                (FSETPC PC PBYTESPERCHAR 2)
                (FSETPC PC PBYTELEN 2)
          else (FSETPC PC PTYPE THINSTRING.PTYPE)
                (FSETPC PC PBINABLE T)
                (FSETPC PC PBYTESPERCHAR 1)
                (FSETPC PC PBYTELEN 1)))
(FSETPC PC PPOS NIL)
(CL:WHEN NEWCHARLOOKS
 (FSETPC PC PLOOKS (CL:IF (FONTP NEWCHARLOOKS)
 (\TEDIT.UNIQIFY.CHARLOOKS (\TEDIT.CHARLOOKS.FROM.FONT NEWCHARLOOKS)
 TEXTOBJ
 NEWCHARLOOKS))))
(CL:WHEN PARALAST (FSETPC PC PPARALAST T))
(\TEDIT.HISTORYADD TEXTOBJ (\TEDIT.HISTORY.EVENT TEXTOBJ ;ReplaceCode N NIL NIL NIL OLDCHAR))
(CL:UNLESS (OR DONTDISPLAY (NOT (\TEDIT.PRIMARYPANE TEXTOBJ)))
 (\TEDIT.UPDATE.LINES TEXTOBJ 'CHANGED N 1))
(RETURN TSTREAM])
```

(TEDIT.NTHCHARCODE

[LAMBDA (TSTREAM N)

; Edited 7-Jul-2024 11:09 by rmk
; Edited 29-Apr-2024 13:06 by rmk
; Edited 17-Mar-2024 00:27 by rmk
; Edited 1-Feb-2024 09:50 by rmk
; Edited 8-Nov-2023 08:41 by rmk
; Edited 4-Nov-2023 15:23 by rmk

:: Returns the Nth character of TSTREAM. First character is N=1, NIL if out of bounds. If TSTREAM is a selection, treats it as a substring, N is relative to that.

:: This is a user-entry. If TSTREAM is a selection with a SELTEXTSTREAM, it is an external selection not held by the stream. No calls with internal selections should come through here.

```
(LET (TEXTOBJ START-OF-PIECE)
 (DECLARE (SPECVARS START-OF-PIECE))
 [if (type? SELECTION TSTREAM)
  then ; This case should only be a user-entry convenience--internally the stream is always passed.
    (SETQ TEXTOBJ (TEXTOBJ (FGETSEL TSTREAM SELTEXTSTREAM)))
    (CL:UNLESS (EQ N 0)
      [add N (CL:IF (ILESSP N 0)
                    (FGETSEL TSTREAM CHLIM)
                    (SUB1 (FGETSEL TSTREAM CH#)))]
      (CL:WHEN (OR (ILESSP N (FGETSEL TSTREAM CH#))
                  (IGEQ N (FGETSEL TSTREAM CHLIM)))
        ; Out of the selection: force NIL
        (SETQ N 0)))
  else (SETQ TEXTOBJ (TEXTOBJ TSTREAM))
        (CL:WHEN (ILESSP N 0)
          (SETQ N (IPLUS (FGETTOBJ TEXTOBJ TEXTLEN)
                        N 1)))]
(CL:WHEN (AND (IGEQ N 1)
              (ILEQ N (FGETTOBJ TEXTOBJ TEXTLEN)))
 (\TEDIT.PIECE.NTHCHARCODE TEXTOBJ (\TEDIT.CHTOPC N TEXTOBJ T)
 (IDIFFERENCE (ADD1 N)
 START-OF-PIECE))))]
```

(TEDIT.NTHCHAR

[LAMBDA (TSTREAM N)

; Edited 22-Oct-2024 21:28 by rmk

```
(LET ((CODE (\TEDIT.NTHCHARCODE TSTREAM N)))
 (CL:IF (FIXP CODE)
 (CHARACTER CODE)
 CODE)])
```

(\TEDIT.PIECE.NTHCHARCODE

[LAMBDA (TEXTOBJ PC OFFSET)

; Edited 21-Oct-2024 00:26 by rmk
; Edited 29-Apr-2024 08:46 by rmk
; Edited 22-Mar-2024 00:02 by rmk
; Edited 1-Feb-2024 09:55 by rmk
; Edited 6-Jan-2024 16:36 by rmk
; Edited 29-Dec-2023 11:55 by rmk
; Edited 8-Dec-2023 22:54 by rmk
; Edited 7-Dec-2023 15:57 by rmk

; Edited 8-Nov-2023 08:43 by rmk
; Edited 5-Nov-2023 08:17 by rmk

:: Returns the OFFSETth charcode of PC, NIL if OFFSET is out of bounds. For file pieces, ensures that the backing stream is restored to its original
:: position, so that it remains compatible with the values (buffer, offset) in the textstream.

```
(CL:WHEN (AND (IGEQ OFFSET 1)
              (ILEQ OFFSET (PLEN PC)))
  [LET ((PCONTENTS (PCONTENTS PC))
        FILEPOS)
      (SELECTC (PTYPE PC)
        (STRING.PTYPE
          (NTHCHARCODE PCONTENTS OFFSET))
        (THINFILE.PTYPE
          (SETQ FILEPOS (\GETFILEPTR PCONTENTS))
          (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                         (SUB1 OFFSET)))
          (PROG1 (BIN PCONTENTS)
                (\SETFILEPTR PCONTENTS FILEPOS)))
        (FATFILE1.PTYPE
          (SETQ FILEPOS (\GETFILEPTR PCONTENTS))
          (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                         (SUB1 OFFSET)))
          (PROG1 (create WORD
                    HIBYTE _ (PCHARSET PC)
                    LOBYTE _ (BIN PCONTENTS))
                (\SETFILEPTR PCONTENTS FILEPOS)))
        (FATFILE2.PTYPE
          (SETQ FILEPOS (\GETFILEPTR PCONTENTS))
          (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                         (UNFOLD (SUB1 OFFSET)
                                                2)))
          (PROG1 (\WIN PCONTENTS)
                (\SETFILEPTR PCONTENTS FILEPOS)))
        (UTF8.PTYPE (SETQ FILEPOS (\GETFILEPTR PCONTENTS))
                    [\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                                    (ITIMES (SUB1 OFFSET)
                                                            (PBYTESPERCHAR PC)
                                                            (PROG1 (UTF8.INCCODEFN PCONTENTS)
                                                                (\SETFILEPTR PCONTENTS FILEPOS)))
                    (OBJECT.PTYPE PCONTENTS)
                    (SUBSTREAM.PTYPE ; A substream stored as an object
                     (\TEDIT.THELP 'SUBSTREAM?)
                     (BIN (IMAGEOBJPROP PCONTENTS 'SUBSTREAM)))
                    (PROGN ; For pieces not listed because they require more work. Assumes the function updates COFFSET and that multi-byte
                          ; characters are safe: don't cross buffer boundaries.
                     (\TEDIT.THELP '\TEDIT.PIECE.NTHCHARCODE)]))
      )
    )
```

:: Slashed functions. Private?

(DEFINEQ

(\TEDIT1

[LAMBDA (TSTREAM WINDOW UNSPAWNED)

; Edited 24-Apr-2024 10:38 by rmk
; Edited 17-Mar-2024 12:51 by rmk
; Edited 22-Sep-2023 20:23 by rmk
; Edited 13-Sep-2023 22:37 by rmk
; Edited 12-Jun-90 17:51 by mitani

:: Does the actual editing work, once TEDIT has OPENTEXTSTREAMed the thing to be edited.

```
(LET ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
      (\TEDIT.COMMAND.LOOP TSTREAM)
      (CLOSEW WINDOW)
      (\TEDIT.TEXTCLOSEF TSTREAM)
      (replace (STREAM ACCESSBITS) of TSTREAM with BothBits)
      (CL:WHEN (GETTEXTPROP TEXTOBJ 'AFTERQUIIFN)
        (APPLY* (GETTEXTPROP TEXTOBJ 'AFTERQUIIFN)
                WINDOW TSTREAM))
      (CL:WHEN UNSPAWNED)
      (COND
        ((NEQ T (FGETTOBJ TEXTOBJ EDITFINISHEDFLG))
         (PROG1 (FGETTOBJ TEXTOBJ EDITFINISHEDFLG)
                (FSETTOBJ TEXTOBJ EDITFINISHEDFLG NIL)))
        ((STRINGP (FGETTOBJ TEXTOBJ TXTFILE))
         (COERCETEXTOBJ TEXTOBJ 'STRINGP))
        (T TSTREAM))))]
```

; Run the editing engine
; Close the edit window
; Close the underlying files
; But leave the stream itself accessible
; Apply any post-window-close (and post-QUIT) function

; We're not a distinct process: Send back the edited text in some
; suitable form

(\TEDIT.INSERT

[LAMBDA (INSERT SEL TSTREAM DONTSCROLL TYPEIN)

; Edited 5-Jan-2025 23:01 by rmk
; Edited 28-Nov-2024 09:53 by rmk
; Edited 25-Nov-2024 22:05 by rmk
; Edited 18-Nov-2024 15:53 by rmk
; Edited 15-Nov-2024 18:05 by rmk


```

; Edited 10-Nov-2024 23:38 by rmk
; Edited 30-Oct-2024 14:49 by rmk
; Edited 18-Oct-2024 22:07 by rmk
; Edited 27-Aug-2024 14:31 by rmk
; Edited 30-Jul-2024 23:55 by rmk
; Edited 21-Jun-2024 23:09 by rmk
; Edited 6-May-2024 13:47 by rmk
; Edited 21-Apr-2024 20:24 by rmk
; Edited 9-Mar-2024 11:36 by rmk
; Edited 14-Jan-2024 12:14 by rmk
; Edited 20-Dec-2023 15:27 by rmk
; Edited 18-Sep-2023 12:48 by rmk
; Edited 24-May-2023 14:23 by rmk
; Edited 29-May-91 18:22 by jds

```

```

;; The selection runs from CH# to CHLIM-1. The insertion is before the selection if POINT=LEFT or after the selection (POINT=RIGHT). This
;; translates to before CH# or before CHLIM respectively.
;; Inserts INSERT at the location picked out by the selection, and then implements all the consequences for line and screen updates. Assumes that
;; the caller got the selection and the text set up properly.
;;
;; Text can be a string or a single charcode (only on the call from \TEDIT.COMMAND.LOOP).

```

```

(CL:WHEN [AND (GETSEL SEL SET)
              (OR (CHARCODEP INSERT)
                  (NEQ 0 (NCHARS INSERT)
                        [PROG* ((TEXTOBJ (fetch TEXTSTREAM TEXTOBJ) of TSTREAM))
                              (PARACHARS (FGETTOBJ TEXTOBJ PARABREAKCHARS))
                              NCHARSADDED CARETCHNO])

```

```

;; FORM is not included in the EOL characters. It breaks a line, but the first line after it doesn't necessarily have first-line margins. CR/LF
;; maybe should have been converted by plaintext reader or \TEXTBOUT.

```

```

(CL:WHEN (\TEDIT.READONLY TEXTOBJ NIL CARETCHNO)
  (RETURN NIL))
(CL:WHEN (FGETTOBJ TEXTOBJ BLUEPENDINGDELETE) ; Blue pending delete?
  (\TEDIT.DELETE TEXTOBJ SEL))
(SETQ CARETCHNO (TEDIT.GETPOINT TEXTOBJ SEL))
(if (CHARCODEP INSERT)
  then ;; Meta,EOL causes a line break but not a paragraph break, in terms of formatting. Original code converted to a
        ;; formatted file on the first appearance of Meta,EOL, not clear why and so removed.
        (\TEDIT.INSERTCH INSERT CARETCHNO TEXTOBJ (FMEMB INSERT PARACHARS))
        (SETQ NCHARSADDED 1)
  elseif (AND PARACHARS (thereis CH instring INSERT suchthat (FMEMB CH PARACHARS)))
  then ;; It's maybe worth a scan here to see if we can insert the string. This avoids the heavier per-character complexity of
        ;; \INSERTCH.
        (for CH instring INSERT as NCH# from CARETCHNO do (\TEDIT.INSERTCH CH NCH# TEXTOBJ
                                                           (MEMB CH PARACHARS)))
        (SETQ NCHARSADDED (NCHARS INSERT))
  else (\TEDIT.INSERTCH INSERT CARETCHNO TEXTOBJ
        (SETQ NCHARSADDED (NCHARS INSERT)))
(FSETTOBJ TEXTOBJ \DIRTY T)

```

```

;;
;; The piece table is now correct: NCHARSADDED new characters have been added in front of CARETCHNO.

```

```

(\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
;; Set the caret so that the next insertion should also come in front of that (now displaced) character, and then update the screen.
;; If typein, the new selection is a point selection, if from a function e.g. TEDIT.INSERT, the insertion is selected/underlined. TEDIT.INSERT
;; can then apply the looks, if specified.

```

```

(if TYPEIN
  then (\TEDIT.UPDATE.SEL SEL (SUB1 (IPLUS CARETCHNO NCHARSADDED))
        0
        'RIGHT
        'NORMAL)
  else (\TEDIT.UPDATE.SEL SEL CARETCHNO NCHARSADDED 'RIGHT 'NORMAL))
(CL:UNLESS DONTSCROLL

```

```

;; All the panes must be updated. SELPANE mayalso need to be scrolled to make the caret visible for the next input.

```

```

(\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION CARETCHNO NCHARSADDED)
(CL:WHEN (EQ SEL (TEXTSEL TEXTOBJ))
  (\TEDIT.SHOWSEL SEL T TEXTOBJ))
(CL:WHEN TYPEIN (\TEDIT.SCROLL.CARET TSTREAM)))]])

```

(\TEDIT.MOVE

```

[LAMBDA (FROMSEL TOSEL FROMTSTREAM TOTSTREAM)

```

```

; Edited 6-Feb-2025 16:17 by rmk
; Edited 8-Dec-2024 21:37 by rmk
; Edited 26-Nov-2024 22:34 by rmk
; Edited 22-Nov-2024 15:42 by rmk
; Edited 22-Sep-2024 18:43 by rmk
; Edited 13-Sep-2024 22:31 by rmk
; Edited 27-Aug-2024 14:27 by rmk
; Edited 7-Jul-2024 16:38 by rmk
; Edited 3-Jul-2024 10:11 by rmk
; Edited 18-May-2024 16:24 by rmk

```

; Edited 29-Apr-2024 12:53 by rmk
; Edited 22-Apr-2024 23:55 by rmk
; Edited 15-Mar-2024 13:54 by rmk
; Edited 12-Oct-2023 22:23 by rmk
; Edited 24-Sep-2023 21:43 by rmk
; Edited 21-Jun-2023 15:58 by rmk
; Edited 29-May-91 18:21 by jds

(SELECTION! FROMSEL)
(SELECTION! TOSEL)

:: Extracts the FROM-selected text from its document and inserts it at the TO-selected position in its document. TOSEL is the SEL of , FROM
:: should be a scratch selection.
:: from FROM and inserts them Insert the pieces at FROM into the location described by TO, possibly first deleting a TO-pending-delete and also
:: removing the FROM pieces.
:: This results in a single history event if FROM and TO are in the same TEXTOBJ--undo will restore it.
:: If they are in separate texts, then the modifications to TO go into TO's history (pending delete and insert), the deletion of FROM is an event in its
:: object. In that case it will require undos in both objects to get them both back to the original state.

(CL:UNLESS (EQ 0 (GETSEL FROMSEL DCH))
[PROG* ((FROMOBJ (GETTSTR FROMSTREAM TEXTOBJ))
(TOOBJ (GETTSTR TOTSTREAM TEXTOBJ))
(TOCH# (FGETSEL TOSEL CH#))
(TODCH (FGETSEL TOSEL DCH))
(TOPOINT (FGETSEL TOSEL POINT))
TODELEVENT FROMPIECES BPD)
(CL:WHEN (\TEDIT.MOVE.MSG FROMOBJ TOOBJ NIL)
(RETURN NIL))

:: TOSEL is the SEL of TOOBJ. Its highlighting is taken down, the insertion happens at that position (maybe after a bluepending deletion. At
:: the end the inserted material is highlighted with caret on the right.
:: FROMSEL is a selection maybe in a different document. Either way, its temporary highlighting is taken down and the selection is deleted.
:: If FROM is in a different document, the Venue sysout leaves that documents SEL as it was before (i.e. not at the position of the deletion).
:: Maybe it should be moved (and scrolled) to a point selection at the deletion site?

:: We'd like to check all the image objects for allowance before we make any changes, but we don't know whether the imageobject
:: WHENxxx functions have side effects even if they decline. So we check only once, when we do the BPD operation or grab the from
:: pieces. Since we are testing for MOVE, presumably the insert and delete are both OK.

:: Get rid of the BPD

(CL:WHEN (AND (FGETTOBJ TOOBJ BLUEPENDINGDELETE)
(IGREATERP TODCH 0))
(FSETTOBJ TOOBJ BLUEPENDINGDELETE NIL)
(CL:UNLESS (\TEDIT.DELETE TOOBJ TOSEL)
(RETURN NIL))
(SETQ BPD T)
(CL:WHEN (EQ TOOBJ FROMOBJ) ; Same text, pre-adjust the source
(\TEDIT.SEL.DELETEDCHARS FROMSEL TOCH# TODCH)))

:: BPD is gone, TOSEL and FROMSEL have been adjusted

:: Grab (a copy of) the source pieces, if image objects allow copying. FROMPIECES is essentially a clipboard for extract/insert--the
:: FROMOBJ has not yet been changed.

(SETQ FROMPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES FROMSEL NIL FROMOBJ)
'MOVE TOOBJ FROMOBJ))

(CL:UNLESS FROMPIECES
; If bailing, should we undo the BPDEVENT (if history is ON)?
(RETURN))
(\TEDIT.SHOWSEL FROMSEL NIL FROMOBJ)
(\TEDIT.SHOWSEL TOSEL NIL TOOBJ)

:: No need to recheck allowance

(if (EQ TOOBJ FROMOBJ)
then ; Can't call \TEDIT.DELETE because we don't want to implicitly update the TOSEL for the insert.
(\TEDIT.DELETE.SELPIECES FROMOBJ FROMSEL NIL T)
(\TEDIT.SEL.DELETEDCHARS TOSEL FROMSEL)
(\TEDIT.UPDATE.LINES FROMOBJ 'DELETION FROMSEL)
; Pop to accumulate into a single event (BPD, DELETE, INSERT).
else (\TEDIT.DELETE FROMOBJ FROMSEL NIL NIL T))

:: Deletion accomplished possibly in separate FROMOBJ with its own history.

(CL:WHEN (GETTEXTPROP TOOBJ 'COPYBYBKSYSEBUF)
(\TEDIT.FOREIGN.COPY (WFROMDS TOTSTREAM
FROMSEL T)
(CL:WHEN BPD ; If no BPD, TO history is good
(\TEDIT.HISTORYADD.COMPOSITE TOOBJ TOOBJ (LIST (\TEDIT.POPEVENT TOOBJ)
(\TEDIT.POPEVENT TOOBJ))))
(RETURN))
(\TEDIT.INSERT.SELPIECES FROMPIECES TOOBJ TOSEL)

```
(\TEDIT.SET.SEL.LOOKS TOSEL 'NORMAL)
(\TEDIT.FIXSEL TOSEL TOOBJ)
(\TEDIT.SHOWSEL TOSEL T TOOBJ)
```

;;
;; TO history in order has INS, DEL if TO=FROM, and possibly BPD. (DEL) (BPD), put them all in a composite event.

```
(\TEDIT.HISTORYADD.COMPOSITE TOOBJ (LIST (\TEDIT.POPEVENT TOOBJ)
(CL:IF (EQ TOOBJ FROMOBJ)
(\TEDIT.POPEVENT TOOBJ))
(CL:IF BPD (\TEDIT.POPEVENT TOOBJ])))
```

(\TEDIT.COPY

```
[LAMBDA (FROMSEL TOSEL FROMSTREAM TOTSTREAM)
```

```
; Edited 23-Nov-2024 22:45 by rmk
; Edited 22-Nov-2024 15:44 by rmk
; Edited 13-Sep-2024 22:28 by rmk
; Edited 27-Aug-2024 13:37 by rmk
; Edited 24-Aug-2024 00:17 by rmk
; Edited 7-Jul-2024 22:04 by rmk
; Edited 2-Jul-2024 10:40 by rmk
; Edited 18-May-2024 16:21 by rmk
; Edited 12-May-2024 20:54 by rmk
; Edited 22-Apr-2024 23:55 by rmk
; Edited 29-Apr-2024 12:54 by rmk
; Edited 15-Mar-2024 13:54 by rmk
; Edited 20-Feb-2024 17:03 by rmk
; Edited 1-Feb-2024 20:37 by rmk
; Edited 20-May-2023 18:47 by rmk
; Edited 15-May-2023 22:11 by rmk
; Edited 4-Jun-92 11:11 by jds
```

;; Copy the FROM-selected pieces into the destination object and position as indicated by the TO selection. These may be internal or external, but
;; either way the strams are passed separately.
;; This results in a single history event, either :Insert or :Replace depending on bluependingdelete.

```
(CL:UNLESS (ZEROP (GETSEL FROMSEL DCH))
(PROG* ((FROMOBJ (GETTSTR FROMSTREAM TEXTOBJ))
(TOOBJ (GETTSTR TOTSTREAM TEXTOBJ))
FROMPIECES)
(CL:WHEN (\TEDIT.MOVE.MSG FROMOBJ TOOBJ T)
(RETURN))
```

;; Grab (a copy of) the source pieces, if image object allows

```
(SETQ FROMPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES FROMSEL NIL FROMOBJ)
'COPY TOOBJ FROMOBJ))
(CL:UNLESS FROMPIECES (RETURN))
```

;; No object objected

```
(\TEDIT.SHOWSEL FROMSEL NIL FROMOBJ) ; Turn off any current highlighting
(\TEDIT.SHOWSEL TOSEL NIL TOOBJ)
```

;;

```
(CL:WHEN (GETTEXTPROP TOOBJ 'COPYBYBKSYSEBUF)
(\TEDIT.FOREIGN.COPY (WFROMDS TOTSTREAM)
FROMSEL T)
(RETURN))
```

;;

;; Install FROM pieces at TO, either replacing or inserting depending on its BLUEPENDINGDELETE

```
(if (FGETTOBJ TOOBJ BLUEPENDINGDELETE)
then (FSETTOBJ TOOBJ BLUEPENDINGDELETE NIL)
(\TEDIT.REPLACE.SELPIECES FROMPIECES TOOBJ TOSEL)
else (\TEDIT.INSERT.SELPIECES FROMPIECES TOOBJ TOSEL))
```

;;

```
(\TEDIT.SET.SEL.LOOKS TOSEL 'NORMAL)
```

;;

```
(\TEDIT.SHOWSEL TOSEL NIL TOOBJ) ; Take down anything that might have appeared
(\TEDIT.FIXSEL TOSEL TOOBJ)
(\TEDIT.SCROLL.CARET TOTSTREAM)))
```

(\TEDIT.REPLACE.SELPIECES

```
[LAMBDA (INSERTSELPIECES TEXTOBJ SEL)
```

```
; Edited 8-Dec-2024 13:46 by rmk
; Edited 26-Nov-2024 17:37 by rmk
; Edited 29-Sep-2024 00:24 by rmk
; Edited 21-Sep-2024 22:12 by rmk
; Edited 13-Sep-2024 22:28 by rmk
; Edited 7-Jul-2024 11:52 by rmk
; Edited 5-Jul-2024 23:21 by rmk
; Edited 18-May-2024 16:47 by rmk
; Edited 12-May-2024 21:13 by rmk
; Edited 17-Feb-2024 16:34 by rmk
; Edited 15-Mar-2024 13:32 by rmk
; Edited 27-May-2023 11:22 by rmk
; Edited 24-May-2023 22:38 by rmk
```

;; Replaces the selection SEL of TEXTOBJ with INSERTSELPIECES. Produces a :Replace history event. TEXTOBJ will remember the insertion
 ;; for undoing, our history event only has to keep track of the pieces it replaced, and where they were.
 ;; If SEL includes an object that declines deletion, the document is left unchanged.
 ;; On return, the pieces, lines, selection, and display are complete, correct, and consistent

```
(CL:UNLESS (\TEDIT.READONLY TEXTOBJ)
  (PROG ((POINT (GETSEL SEL POINT))
    (CH# (FGETSEL SEL CH#))
    (DCH (FGETSEL SEL DCH))
    (ILEN (GETSPC INSERTSELPIECES SPLEN))
    DELEVENT)
  (\TEDIT.SHOWSEL SEL NIL TEXTOBJ)
```

;; We first delete, then insert, updating the display after the second operation.

```
(CL:WHEN (\TEDIT.DELETE.SELPIECES TEXTOBJ CH# DCH)
  ;; Reduce to a point to the right of the last remaining character so that FIXSEL sees starting character in its proper line.
  (\TEDIT.UPDATE.SEL SEL (SUB1 CH#)
    0
    'RIGHT
    'NORMAL)
  (\TEDIT.FIXSEL SEL TEXTOBJ)
  (FSETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ SEL))
  (SETQ DELEVENT (\TEDIT.LASTEVENT TEXTOBJ T)) ; Catch the deletion event
```

;;

```
(CL:WHEN (AND (IGEQ ILEN 0)
  (\TEDIT.INSERT.SELPIECES INSERTSELPIECES TEXTOBJ SEL T))
```

;; If both delete and insert happened, foush the insert event and upgrade the DELEVENT to a single :Replace. The insert has
 ;; not updated the lines or the selection

```
(CL:WHEN DELEVENT
  (\TEDIT.POPEVENT TEXTOBJ)
  (SETTH DELEVENT THACTION :Replace)
  (SETTH DELEVENT THLEN ILEN)
  (SETTH DELEVENT THPOINT POINT))
  (\TEDIT.UPDATE.SEL SEL CH# ILEN POINT))
  (if (IGREATERP ILEN DCH)
    then (\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION CH# (IDIFFERENCE ILEN DCH))
    elseif (ILESSP ILEN DCH)
    then (\TEDIT.UPDATE.LINES TEXTOBJ 'DELETION CH# (IDIFFERENCE DCH ILEN))
    else (\TEDIT.UPDATE.LINES TEXTOBJ 'LOOKS CH# DCH))))
```

(\TEDIT.INSERT.SELPIECES

```
[LAMBDA (SELPIECES TEXTOBJ TARGETSEL DONTUPDATE)
```

; Edited 26-Nov-2024 11:04 by rmk
 ; Edited 31-Oct-2024 18:01 by rmk
 ; Edited 22-Sep-2024 18:37 by rmk
 ; Edited 15-Aug-2024 10:49 by rmk
 ; Edited 5-Jul-2024 23:22 by rmk
 ; Edited 17-Mar-2024 11:43 by rmk
 ; Edited 15-Feb-2024 23:58 by rmk
 ; Edited 13-Feb-2024 09:01 by rmk
 ; Edited 11-Feb-2024 11:42 by rmk
 ; Edited 29-Aug-2023 10:35 by rmk
 ; Edited 12-Aug-2023 11:57 by rmk
 ; Edited 17-Jun-2023 17:08 by rmk
 ; Edited 2-Jun-2023 12:02 by rmk
 ; Edited 31-May-2023 09:56 by rmk
 ; Edited 21-May-2023 23:57 by rmk

;; Insert SELPIECES into TEXTOBJ at TARGETSEL's caret. TARGETSEL can be a character position.
 ;; \TEDIT.INSERTCH.HISTORY uses the first piece to decide whether it is in a consecutive run of insertions.

```
(CL:WHEN SELPIECES
  (LET ((INSCH# (TEDIT.GETPOINT TEXTOBJ TARGETSEL))
    (SPLEN (fetch (SELPIECES SPLEN) of SELPIECES))
    (SPFIRST (fetch (SELPIECES SPFIRST) of SELPIECES))
    NEXTPC)
  (SETQ NEXTPC (\TEDIT.ALIGNEDPIECE INSCH# TEXTOBJ))
  (\TEDIT.INSERTPIECES SPFIRST NEXTPC TEXTOBJ)
  (\TEDIT.DIFFUSE.PARALOOKS (PREVPIECE SPFIRST)
    NEXTPC)
  (CL:UNLESS DONTUPDATE ; Under replace?
    ;; Adjust SEL to select the inserted material, with point on the right.
    (\TEDIT.UPDATE.SEL (FGETTOBJ TEXTOBJ SEL)
      INSCH# SPLEN 'RIGHT)
    (\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION INSCH# SPLEN))
  (\TEDIT.HISTORYADD TEXTOBJ (\TEDIT.HISTORY.EVENT TEXTOBJ :Insert INSCH# SPLEN NIL SPFIRST))))
```

(\TEDIT.RESTARTFN

```
[LAMBDA (TSTREAM WINDOW PROPS)
```

; Edited 29-Jun-2024 00:02 by rmk
 ; Edited 24-Apr-2024 10:38 by rmk
 ; Edited 17-Mar-2024 16:58 by rmk
 ; Edited 22-Sep-2023 20:31 by rmk

; Edited 21-Aug-2022 08:13 by rmk
; Edited 12-Jun-90 17:51 by mitani

:: RMK: not sure why it needs to fool around with the window and create a new stream. Why not just restart the process and command loop?

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(SETQ TSTREAM (OPENTEXTSTREAM TSTREAM WINDOW NIL NIL PROPS))
(LET* ((TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
      (ODIRTY (GETTOBJ TEXTOBJ \DIRTY)))
  (SETTOBJ TEXTOBJ PRIMARYPANE)
  (SETQ TSTREAM (OPENTEXTSTREAM TSTREAM WINDOW NIL NIL PROPS))
  (SETQ TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM))
  (SETTOBJ TEXTOBJ \DIRTY ODIRTY)
  (\TEDIT.COMMAND.LOOP TSTREAM)
  (CLOSEW WINDOW)
  (\TEDIT.TEXTCLOSEF TSTREAM)
  (replace (STREAM ACCESSBITS) of TSTREAM with BothBits)
  (CL:WHEN (GETTEXTPROP TEXTOBJ 'AFTERQUITFN)
    (APPLY* (GETTEXTPROP TEXTOBJ 'AFTERQUITFN)
      WINDOW TSTREAM)))
```

; Unattach the window, so we don't do a redisplay.
; Reopen, reattach
; New stream maybe new textobj.
; Preserve dirty
; Run the editing engine
; Close the edit window. WHY ??
; Close the underlying files
; But leave the stream itself accessible
; Apply any post-window-close (and post-QUIT) function

(\TEDIT.CHARDELETE

[LAMBDA (TSTREAM FORWARD)

; Edited 28-Nov-2024 10:14 by rmk
; Edited 27-Nov-2024 09:18 by rmk
; Edited 29-Sep-2024 21:04 by rmk
; Edited 22-Sep-2024 18:56 by rmk
; Edited 8-Jul-2024 00:12 by rmk
; Edited 23-Jun-2024 19:41 by rmk
; Edited 1-Apr-2024 22:44 by rmk
; Edited 17-Mar-2024 00:27 by rmk
; Edited 23-Dec-2023 17:32 by rmk
; Edited 22-May-2023 23:24 by rmk
; Edited 19-Apr-93 10:50 by jds

:: This identifies the character before or after the current caret position, and deletes it.

```
(CL:UNLESS (\TEDIT.READONLY TSTREAM)
  (PROG* ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
        (SEL (TEXTSEL TEXTOBJ))
        (DCH (GETSEL SEL DCH))
        CH#)
    (if (AND (FGETTOBJ TEXTOBJ BLUEPENDINGDELETE)
            (IGREATERP (GETSEL SEL DCH)
              1))
      then
        ;; If there is a BPD selection, just delete that. If only length 1, it's probably the first backspace after a selection in
        ;; XYTOSEL. Back it up as if was the second backspace (with DCH=0) below.
        (SETQ CH# (FGETSEL SEL CH#))
      elseif (SETQ CH# (CL:IF FORWARD
        (\TEDIT.NEXTCHANGEABLE.CHNO (TEDIT.GETPOINT TSTREAM SEL)
          TEXTOBJ)
        (\TEDIT.LASTCHANGEABLE.CHNO (SUB1 (TEDIT.GETPOINT TSTREAM SEL)
          TEXTOBJ)))
      then
        ;; Target the first visible character before or after, unless it is protected
        (SETQ DCH 1)
      else (RETURN))
    (\TEDIT.DELETE TEXTOBJ CH# DCH (CL:IF FORWARD
      'RIGHT
      'LEFT))
    (\TEDIT.SCROLL.CARET TSTREAM)))
```

(\TEDIT.COPYPIECE

[LAMBDA (PC FROMOBJ TOOBJ UNPROTECT OPERATION PROMPTTEXTOBJ)

; Edited 3-Aug-2024 12:40 by rmk
; Edited 15-Oct-2023 20:14 by rmk
; Edited 30-Jul-2023 22:44 by rmk
; Edited 21-Jun-2023 00:15 by rmk
; Edited 19-May-2023 21:39 by rmk
; Edited 7-May-2023 11:46 by rmk
; Edited 12-Jun-90 17:50 by mitani

:: TEXTOBJ's prompt gets the message that a copy is not allowed, the FROMOBJ and TOOBJ provide the streams for the object's copy function.
:: The copy is disconnected from PC's original connections.

:: If UNPROTECT, the copies of protected pieces are unprotected

:: OPERATION keys which imageobject function to apply, if any

```
(PROG (NEWPC SRCPCFILE (CROSSCOPY (NEQ FROMOBJ TOOBJ)))
  (SETQ NEWPC (create PIECE using PC PNEW _ T PREVPIECE _ NIL NEXTPIECE _ NIL PREENODE _ NIL))
  (SELECTC (PTYPE PC)
    (FILE.PTYPES (CL:WHEN CROSSCOPY
```

; No matter what, we need a fresh copy.

;; If this is a cross-document copy, and the text comes from a file, we must REALLY make a copy of the text,
;; lest the source file be deleted. We want to preserve the external format, so that we can just copy the bytes.

```
(SETQ SRCPCFILE (PCONTENTS PC))
(CL:UNLESS (\GETSTREAM SRCPCFILE 'INPUT T)
  ; The source file was CLOSED -- reopen it, for our use
```

```

[SETQ SRCPPFILE (OPENSTREAM SRCPPFILE 'INPUT 'OLD ' ((TYPE TEXT))
[FSETPC PC PCONTENTS (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW
      \(:EXTERNAL-FORMAT , (STREAMPROP SRCPPFILE
      :EXTERNAL-FORMAT])
[ COPYBYTES SRCPPFILE (PCONTENTS PC)
  (PFPOS PC)
  (IPLUS (PFPOS PC)
    (ITIMES (PLEN PC)
      (PBYTESPERCHAR PC]
(FSETPC PC PFPOS 0)))
(STRING.PTYPES
  (change (PCONTENTS NEWPC)
    (CONCAT DATUM)))
(OBJECT.PTYPE
  ; In case this is in the current insertion string
  ; No copy if object doesn't allow it. Caller must be prepared for
  ; NIL?
  (FSETPC NEWPC PCONTENTS (OR (\TEDIT.APPLY.OBJFN (PCONTENTS NEWPC)
    OPERATION FROMOBJ TOOBJ PROMPTTEXTOBJ)
  (RETURN NIL))))
NIL)

```

:: If moving from one text to another, we have to register the looks.

```

(if (AND UNPROTECT (ffetch CLPROTECTED of (PLOOKS NEWPC)))
  then (FSETPC NEWPC PLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS (create CHARLOOKS
    using (PLOOKS PC)
    CLPROTECTED _ NIL CLSELAFTER _ NIL)
    TOOBJ))
  elseif CROSSCOPY
  then (FSETPC NEWPC PLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS (PLOOKS PC)
    TOOBJ))
    (FSETPC NEWPC PPARALOOKS (\TEDIT.UNIQUIFY.PARALOOKS (PPARALOOKS PC)
    TOOBJ)))
(RETURN NEWPC])

```

(\TEDIT.APPLY.OBJFN

```

[LAMBDA (OBJ OPERATION FROMOBJ TOTOBJ PROMPTTEXTOBJ)
; Edited 25-Jun-2024 11:59 by rmk
; Edited 15-Mar-2024 15:38 by rmk
; Edited 15-Jul-2023 10:43 by rmk
; Edited 9-Jul-2023 16:24 by rmk
; Edited 6-Jun-2023 13:35 by rmk
; Edited 30-May-2023 08:15 by rmk
; Edited 19-May-2023 21:37 by rmk
; Edited 7-May-2023 11:46 by rmk

```

:: As part of an OPERATION on an image object piece, we execute the appropriate object functions. If any of them returns DONT, we print a message in the prompt window of PROMPTTEXTOBJ or FROMOBJ, and return NIL. Otherwise, we return an object, either OBJ or a copy.

```

(CL:UNLESS TOTOBJ (SETQ TOTOBJ FROMOBJ))
(PROG [NEWOBJ (OBJFN (IMAGEOBJPROP OBJ (SELECTQ OPERATION
  (COPY 'COPYFN)
  (MOVE 'WHENMOVEDFN)
  (INSERT 'WHENINSERTEDFN)
  (DELETE 'WHENDELETEDFN)
  ; This may want to apply to the first pane?
  'WHENDELETEDFN)
  NIL])
  (SETQ NEWOBJ (if OBJFN
    then (APPLY* OBJFN OBJ (fetch (TEXTOBJ STREAMHINT) of FROMOBJ)
      (CL:UNLESS (EQ OPERATION 'DELETE)
        (fetch (TEXTOBJ STREAMHINT) of TOTOBJ)))
    elseif (EQ OPERATION 'COPY)
    then (COPYALL OBJ)
    else OBJ))
  (CL:WHEN (MEMB NEWOBJ ' (DON'T DONT))
    (TEDIT.PROMPTPRINT (OR PROMPTTEXTOBJ FROMOBJ)
      (CONCAT (L-CASE OPERATION T)
        " of this object not allowed.")
      T)
    (RETURN))

```

:: The WHENCOPIEDFN wants the display stream, not just the text stream.

```

(CL:WHEN [AND (EQ OPERATION 'COPY)
  (SETQ OBJFN (IMAGEOBJPROP OBJ 'WHENCOPIEDFN))
  (MEMB (APPLY* OBJFN OBJ (WINDOWPROP (\TEDIT.PRIMARYPANE TOTOBJ)
    'DSP)
    (fetch (TEXTOBJ STREAMHINT) of FROMOBJ)
    (fetch (TEXTOBJ STREAMHINT) of TOTOBJ))
    ' (DON'T DONT])
  (RETURN NIL))
(RETURN (OR (IMAGEOBJPROP NEWOBJ)
  OBJ))

```

(\TEDIT.DELETE

```

[LAMBDA (TEXTOBJ TARGETSEL/CHAR LEN POINT DONTCHECK)
; Edited 6-Feb-2025 00:14 by rmk
; Edited 8-Dec-2024 21:39 by rmk
; Edited 28-Nov-2024 10:13 by rmk
; Edited 27-Nov-2024 09:18 by rmk
; Edited 13-Sep-2024 22:30 by rmk

```

; Edited 8-Sep-2024 00:07 by rmk
; Edited 7-Jul-2024 12:07 by rmk
; Edited 23-Jun-2024 19:27 by rmk
; Edited 18-May-2024 16:20 by rmk
; Edited 12-May-2024 20:51 by rmk
; Edited 23-Apr-2024 07:35 by rmk
; Edited 24-Apr-2024 10:42 by rmk
; Edited 15-Mar-2024 13:36 by rmk
; Edited 21-Feb-2024 20:40 by rmk
; Edited 20-Feb-2024 20:09 by rmk
; Edited 19-Feb-2024 11:48 by rmk
; Edited 16-Feb-2024 08:46 by rmk
; Edited 12-Nov-2023 12:14 by rmk
; Edited 29-Oct-2023 00:19 by rmk
; Edited 6-Jun-2023 12:48 by rmk
; Edited 29-May-91 18:22 by jds

:: Delete the DCH characters selected by TARGETSEL in TEXTOBJ. Unlike insert, the initial position of the caret doesn't matter.
:: On return, the pieces, lines, and selection are complete and correct, and the display are correct.
::
:: If this is called as part of a move, SEL should end up at the location of the insert, adjusted if the TARGETSEL comes earlier. If this is just a
:: delete, SEL should end up as a point selection at TARGETSEL's CH#.

```
(CL:UNLESS TARGETSEL/CHAR
 (SETQ TARGETSEL/CHAR (TEXTSEL TEXTOBJ))
(LET ((SEL (TEXTSEL TEXTOBJ))
      CLOOKS FIRSTCHAR)
  [if (type? SELECTION TARGETSEL/CHAR)
    then (SETQ CLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ TARGETSEL/CHAR))
         (SETQ FIRSTCHAR (FGETSEL TARGETSEL/CHAR CH#))
         (CL:UNLESS LEN
          (SETQ LEN (FGETSEL TARGETSEL/CHAR DCH)))
         (SETQ POINT (FGETSEL TARGETSEL/CHAR POINT))
    else (SETQ FIRSTCHAR TARGETSEL/CHAR)
         (CL:UNLESS POINT
          (SETQ POINT 'LEFT))])
[SETQ CLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ (CL:IF (EQ POINT 'LEFT)
                                                         (SUB1 FIRSTCHAR)
                                                         (IPLUS FIRSTCHAR LEN)))]
(CL:WHEN (\TEDIT.DELETE.SELPIECES TEXTOBJ FIRSTCHAR LEN DONTCHECK)
```

; Delete the selected characters (if objects allow)

:: Pieces are gone, make lines, SEL, and caret looks consistent with current text.

```
(\TEDIT.UPDATE.LINES TEXTOBJ 'DELETION FIRSTCHAR LEN)
(FSETTOBJ TEXTOBJ CARETLOOKS CLOOKS)
```

:: Adjust SEL and TARGETSEL to reflect the deleted characters.

```
(\TEDIT.SEL.DELETEDCHARS SEL FIRSTCHAR LEN)
```

:: In any event, TARGETSEL's characters are all gone, reduce it to a point selection in case it is still in use. And then SEL moves to
:: the position of the deletion.

:: This is to the right of the last remaining character so that FIXSEL sees starting character in its proper line.

```
(\TEDIT.UPDATE.SEL SEL (SUB1 FIRSTCHAR)
 0
 'RIGHT)
(\TEDIT.FIXSEL SEL TEXTOBJ)
(\TEDIT.SHOWSEL SEL T TEXTOBJ)
(\TEDIT.RESET.EXTEND.PENDING.DELETE TEXTOBJ
 T))
```

(\TEDIT.DIFFUSE.PARALOOKS

```
[LAMBDA (PRIORPC SUCCEEDINGPC)
```

; Edited 16-Feb-2024 00:07 by rmk
; Edited 1-Jul-2023 19:24 by rmk
; Edited 11-Apr-2023 00:08 by rmk
; Edited 22-Oct-2022 22:22 by rmk
; Edited 5-Sep-2022 14:32 by rmk
; Edited 23-Aug-2022 08:40 by rmk
; Edited 12-Jun-90 17:48 by mitani

:: Given a discontinuity in paragraph looks, caused by an insertion or by a deletion: Diffuse the existing paragraph looks across the discontinuity, so
:: that all the pieces in a single paragraph have consistent paragraph looks. Give preference to diffusion toward the END of the document. This
:: means that if you delete an EOL between paragraphs, the second para is absorbed into the first.
:: PRIORPC and SUCCEEDINGPC are the pieces that bound the area of potential discontinuity: the change will occur at one boundary or the
:: other.

```
(CL:WHEN (AND PRIORPC (NOT (PPARALAST PRIORPC)))
```

; The discontinuity is inside a paragraph. Must copy para looks
; forward into the text.

```
(for PC (PPLOOKS _ (PPARALOOKS PRIORPC))
  inpieces
  (NEXTPIECE PRIORPC) until (PPARALAST PC) do
```

(FSETPC PC PPARALOOKS PPLOOKS)) ; Copy para looks info in from the left, up to the first para break.

```
(CL:WHEN SUCCEEDINGPC
 (for PC (PPLOOKS _ (PPARALOOKS SUCCEEDINGPC))
  backpieces
  (PREVPIECE SUCCEEDINGPC) until (OR (EQ PC PRIORPC)
                                       (PPARALAST PC))
```

; Copy para looks back from the right, up to the first para break

do (FSETPC PC PPARALOOKS PPLOOKS)))]

(\TEDIT.WORDDELETE

[LAMBDA (TSTREAM)

; Edited 27-Nov-2024 23:21 by rmk
; Edited 31-Oct-2024 17:47 by rmk
; Edited 7-Jul-2024 11:35 by rmk
; Edited 29-Apr-2024 11:01 by rmk
; Edited 20-Mar-2024 11:08 by rmk
; Edited 25-Dec-2023 00:03 by rmk
; Edited 23-May-2023 16:37 by rmk
; Edited 22-May-2023 10:52 by rmk
; Edited 29-May-91 18:22 by jds

(LET ((TEXTOBJ (GETTSTR TSTREAM TEXTOBJ))
LASTNO FIRSTNO)

(SETQ LASTNO (SUB1 (TEDIT.GETPOINT TSTREAM)))

:: LASTNO is the final (i.e., highest-numbered) character to be deleted.

(CL:UNLESS (ILEQ LASTNO 0)

; Nothing to delete at start of file.

(SETQ FIRSTNO (\TEDIT.WORD.FIRST TSTREAM LASTNO))

(\TEDIT.DELETE TEXTOBJ FIRSTNO (ADD1 (IDIFFERENCE LASTNO FIRSTNO))))]

(\TEDIT.WORDDELETE.FORWARD

[LAMBDA (TSTREAM)

; Edited 27-Nov-2024 20:31 by rmk
; Edited 31-Oct-2024 17:47 by rmk
; Edited 7-Jul-2024 11:35 by rmk
; Edited 29-Apr-2024 10:59 by rmk
; Edited 20-Mar-2024 10:54 by rmk
; Edited 25-Dec-2023 00:20 by rmk
; Edited 23-May-2023 16:37 by rmk
; Edited 22-May-2023 10:52 by rmk
; Edited 29-May-91 18:22 by jds

:: This deletes all characters from the character just after the caret to the end of the following word, skipping over separators to reach the target
:: word.

(LET ((TEXTOBJ (TEXTOBJ! (GETTSTR TSTREAM TEXTOBJ)))
FIRSTNO LASTNO)

:: LASTNO is the final (i.e., highest-numbered) character to be deleted.

(SETQ FIRSTNO (TEDIT.GETPOINT TSTREAM))

(CL:UNLESS (IGREATERP FIRSTNO (TEXTLEN TEXTOBJ))

; Nothing to delete at end of file.

(SETQ LASTNO (\TEDIT.WORD.LAST TSTREAM FIRSTNO))

(\TEDIT.DELETE TEXTOBJ FIRSTNO (ADD1 (IDIFFERENCE LASTNO FIRSTNO))))]

(\TEDIT.FINISHEDIT?

[LAMBDA (TSTREAM NOFORCE)

; Edited 14-Jul-2024 12:25 by rmk
; Edited 1-Jul-2024 16:11 by rmk
; Edited 30-Jun-2024 12:36 by rmk
; Edited 25-Jun-2024 11:59 by rmk
; Edited 2-May-2024 21:25 by rmk
; Edited 20-Mar-2024 10:53 by rmk
; Edited 15-Mar-2024 15:38 by rmk
; Edited 22-Sep-2023 20:21 by rmk
; Edited 20-Sep-2023 23:24 by rmk
; Edited 12-Jun-90 17:50 by mitani

:: Called to determine whether the edit in TSTREAM can be terminated. If there are no active operations and non of the QUITFNS (if any) returns
:: DON'T, then the stream EDITFINISHEDFLG is set to T and NIL is returned. Setting the flag to T will allow the edit process to terminate.

:: Otherwise, the return value is DON'T, so that this can be used by itself to guard closing as a CLOSEWFN.

:: Menus can always be closed.

(PROG* ((TEXTOBJ TEXTOBJ TSTREAM))

(QUITFNS (GETTEXTPROP TEXTOBJ 'QUITFN))

(PRIMPANE (FGETTOBJ TEXTOBJ PRIMARPANE))

QUITFLG)

(CL:WHEN (FGETTOBJ TEXTOBJ EDITOPACTIVE)

:: We're busy doing something, don't close with a message

(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "Not closed: " (CL:IF (EQ T (FGETTOBJ TEXTOBJ EDITOPACTIVE))

"Edit"

(FGETTOBJ TEXTOBJ EDITOPACTIVE))

" operation in progress")

T)

(RETURN 'DON'T))

[for QUITFN (PRIMPANE _ (FGETTOBJ TEXTOBJ PRIMARPANE)) inside QUITFNS

until (OR (EQ QUITFLG 'DON'T)

(EQ QUITFLG T))

do (SETQ QUITFLG (COND

((EQ QUITFN T))

(QUITFN (APPLY* QUITFN PRIMARPANE TSTREAM TEXTOBJ (FGETTOBJ TEXTOBJ EDITOPACTIVE))

(CL:WHEN (EQ QUITFLG 'DON'T)

:: The user supplied a QUITFN that returned DON'T. Editing continues.

(RETURN 'DON'T))

:: If this document has changed, check with the user to make sure he really wants to do it. The question is suppressed for menus and if the

:: QUITFNs gave us T.

```
(CL:WHEN (AND (FGETTOBJ TEXTOBJ \DIRTY)
              (NOT (FGETTOBJ TEXTOBJ MENUFLG)))
  (CL:UNLESS (AND (FGETTOBJ TEXTOBJ MENUFLG)
                 (EQ QUITFNS T)
                 (EQ QUITFLG T))
    (CL:UNLESS (MOUSECONFIRM "Not saved yet; LEFT to Quit anyway." T (FGETTOBJ TEXTOBJ
                                                                    PROMPTWINDOW))
      (RETURN 'DON'T))))
```

:: OK, we can quit.

```
(CL:WHEN [AND PRIMPANE (NOT NOFORCE)
          (NEQ PRIMPANE (PROCESSPROP (TTY.PROCESS)
                                     'WINDOW]
  (TTY.PROCESS (WINDOWPROP PRIMPANE 'PROCESS)))
(FSETTOBJ TEXTOBJ EDITFINISHEDFLG T)
(RETURN NIL))
```

)

(DEFINEQ

(\TEDIT.THELP

```
[LAMBDA (MESS1 MESS2)
```

```
; Edited 21-Oct-2024 01:00 by rmk
; Edited 3-Oct-2024 22:06 by rmk
```

```
:: This is used to signal an internal problem with Tedit or its datastructures. Tedit developers should set \TEDIT.THELPFLG is set to T to force
:: breaks to happen when something goes wrong. It defaults to NIL so that ordinary users see the error message but don't get a break somewhere
:: in the Tedit basement.
```

```
(CL:IF \TEDIT.THELPFLG
  (HELP MESS1 MESS2)
  (ERROR MESS1 MESS2 T))])
```

)

(RPAQ? \TEDIT.THELPFLG NIL)

(DEFINEQ

(\TEDIT.PARAPIECES

```
[LAMBDA (SEL/FIRSTCHAR LASTCHAR TEXTOBJ)
```

```
; Edited 29-Nov-2024 18:28 by rmk
; Edited 7-Jul-2024 20:59 by rmk
; Edited 29-Apr-2024 13:14 by rmk
; Edited 3-Mar-2024 13:01 by rmk
; Edited 11-Dec-2023 23:03 by rmk
```

```
:: Produces SELPIECES that starts at the first piece of the paragraph containing FIRSTCHAR and runs to the last piece of the paragraph
:: containing LASTCHAR. Since paragraphs are split anyway, this does not require further splitting.
```

```
(LET (FIRSTCHAR FIRST LAST START-OF-PIECE)
  (DECLARE (SPECVARS START-OF-PIECE))
  (if (type? SELECTION SEL/FIRSTCHAR)
    then (SETQ FIRSTCHAR (FGETSEL SEL/FIRSTCHAR CH#))
        (SETQ LASTCHAR (CL:IF (EQ 0 (FGETSEL SEL/FIRSTCHAR DCH))
                              FIRSTCHAR
                              (FGETSEL SEL/FIRSTCHAR CHLAST)))
    elseif (type? TEDITHISTORYEVENT SEL/FIRSTCHAR)
    then (SETQ FIRSTCHAR (GETTH SEL/FIRSTCHAR THCH#))
        [SETQ LASTCHAR (SUB1 (IPLUS FIRSTCHAR (GETTH SEL/FIRSTCHAR THLEN]
    else (SETQ FIRSTCHAR SEL/FIRSTCHAR)
        (CL:UNLESS LASTCHAR (SETQ LASTCHAR FIRSTCHAR)))
  (SETQ FIRST (\TEDIT.PARA.FIRST TEXTOBJ FIRSTCHAR))
  (SETQ LAST (\TEDIT.PARA.LAST TEXTOBJ LASTCHAR))
  (create SELPIECES
    SPFIRST _ (CDR FIRST)
    SPLAST _ (CDR LAST)
    SPLLEN _ (ADD1 (IDIFFERENCE (CAR LAST)
                                (CAR FIRST)))
    SPFIRSTCHAR _ (CAR FIRST)
    SPLASTCHAR _ (CAR LAST]))
```

(\TEDIT.PARA.FIRST

```
[LAMBDA (TEXTOBJ CHNO PROTECTEDNOTOK)
```

```
; Edited 30-Jan-2025 12:02 by rmk
; Edited 11-Jan-2025 00:08 by rmk
; Edited 17-Mar-2024 00:27 by rmk
; Edited 19-Jan-2024 10:10 by rmk
; Edited 26-Dec-2023 09:14 by rmk
; Edited 24-Dec-2023 22:14 by rmk
; Edited 11-Dec-2023 21:52 by rmk
```

```
:: Returns (FIRSTCHARNO . FIRSTPIECE) of the paragraph containing CHNO. FIRSTCHARNO is the firstcharacter of FIRSTPIECE, because
:: paragraphs start on piece boundaries. When PROTECTEDNOTOK, the scan will terminated on a protected piece, even if that isn't the end of the
:: paragraph.
```

```
(if (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
  then (CONS 0)
  else (LET (CHPIECE START-OF-PIECE START)
    (DECLARE (SPECVARS START-OF-PIECE))
```

```
(if (type? SELPIECES CHNO)
  then (SETQ CHPIECE (GETSPC CHNO SPFIRST))
        (SETQ START (GETSPC CHNO SPFIRSTCHAR))
  elseif (type? PIECE CHNO)
  then (SETQ START (\TEDIT.PCTOCH CHNO TEXTOBJ))
        (SETQ CHPIECE CHNO)
  else [SETQ CHNO (CL:IF (type? SELECTION CHNO)
                        (FGETSEL CHNO CH#)
                        (IMAX 0 (IMIN CHNO (TEXTLEN TEXTOBJ))))]
        (SETQ CHPIECE (\TEDIT.CHTOPC CHNO TEXTOBJ T))
        (SETQ START START-OF-PIECE))
;; Start one before CHPIECE, its PARALAST doesn't matter. Assume CHPIECE is visible
(for PC (PLENTOT _ 0)
  backpieces
  (AND CHPIECE (PREVPIECE CHPIECE)) when (VISIBLEPIECEP PC) until (PPARALAST PC)
  until (AND PROTECTEDNOTOK (GETCLOOKS (PLOOKS PC)
                                        CLPROTECTED))
  do (add PLENTOT (PLEN PC)) finally
    ;; If the iteration reached the beginning, there is no PREVPIECE. Otherwise, PC is the
    ;; previous PARALAST, and we have to take its next
    (RETURN (CONS (IDIFFERENCE START PLENTOT)
                  (CL:IF PC
                       (NEXTPIECE PC)
                       (\TEDIT.FIRSTPIECE TEXTOBJ)))]))
```

(\TEDIT.PARA.LAST

[LAMBDA (TEXTOBJ CHNO PROTECTEDNOTOK)

; Edited 7-Feb-2025 08:32 by rmk
 ; Edited 31-Jan-2025 09:33 by rmk
 ; Edited 17-Mar-2024 00:27 by rmk
 ; Edited 19-Jan-2024 10:37 by rmk
 ; Edited 26-Dec-2023 09:14 by rmk
 ; Edited 24-Dec-2023 22:16 by rmk
 ; Edited 11-Dec-2023 23:02 by rmk

;; Returns (LASTCHARNO . LASTPIECE) of the paragraph containing CHNO. If CHNO is SELPIECES or SELECTION, CHNO is taken as its last
 ;; character. LASTCHARNO is the number of the last character of the paragraph (presumably on EOL). It is also the character of LASTPIECE,
 ;; because paragraphs end on piece boundaries. When PROTECTEDNOTOK, the scan will be terminated on a protected piece, even if that isn't the
 ;; beginning of the paragraph.

```
(if (ZEROP (FGETTOBJ TEXTOBJ TEXTLEN))
  then ; Empty document
  (CONS 0)
  else (LET (CHPIECE START-OF-PIECE END FORMATTED)
    (DECLARE (SPECVARS START-OF-PIECE))
    (if (type? SELPIECES CHNO)
      then (SETQ CHPIECE (GETSPC CHNO SPLAST))
            [SETQ END (SUB1 (IDIFFERENCE (GETSPC CHNO SPLASTCHAR)
                                         (PLEN CHPIECE))
                                (TEXTLEN TEXTOBJ))]
      elseif (type? PIECE CHNO)
      then (SETQ CHPIECE CHNO)
            (SETQ END (\TEDIT.PCTOCH CHNO TEXTOBJ))
      else (SETQ CHPIECE (\TEDIT.CHTOPC (IMIN (CL:IF (type? SELECTION CHNO)
                                                    (FGETSEL CHNO CHLAST)
                                                    CHNO)
                                         (TEXTLEN TEXTOBJ))
                                         TEXTOBJ T))
            (SETQ END START-OF-PIECE))
    ;; END is now the first character of the piece containing CHNO
    (for PC (PLENTOT _ 0)
      inpieces CHPIECE when (VISIBLEPIECEP PC) do (add PLENTOT (PLEN PC))
      repeatuntil (PPARALAST PC) repeatuntil (AND PROTECTEDNOTOK (FGETCLOOKS (PLOOKS PC)
                                                                                CLPROTECTED))
      finally (RETURN (CONS (IMIN (IPLUS END PLENTOT -1)
                                (FGETTOBJ TEXTOBJ TEXTLEN))
                            PC]))
```

)

(DEFINEQ

(\TEDIT.WORD.FIRST

[LAMBDA (TSTREAM CHNO WORDBOUNDTABLE)

; Edited 20-Dec-2024 07:51 by rmk
 ; Edited 29-Apr-2024 10:56 by rmk
 ; Edited 20-Mar-2024 10:54 by rmk
 ; Edited 17-Mar-2024 12:05 by rmk
 ; Edited 25-Dec-2023 18:53 by rmk
 ; Edited 23-May-2023 16:37 by rmk
 ; Edited 22-May-2023 10:52 by rmk
 ; Edited 29-May-91 18:22 by jds

;; Returns the number of the first character of the word containing CHNO or of the word preceding CHNO if CHNO does not map to a text
 ;; character. Unlike the paragraph case, we don't get much help from the pieces, because words are not piece-aligned. Caller can do the piece
 ;; manipulation given the result.

;; We don't need to worry about invisibles here, \BACKBIN skips them.

;; Image objects are treated as text characters.

;; Punctuation is tricky: It stops whitespace and text, and its immediate predecessor doesn't matter.

```
(if (ILEQ CHNO 1)
  then 1
  else (PROG ((TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))
             (READSA CH)
             (SETQ READSA (fetch READSA of (OR WORDBOUNDTABLE (FGETTOBJ TEXTOBJ TXTWTBL)
                                                    TEDIT.WORDBOUND.READTABLE)))
             (SETQ CHNO (IMIN CHNO (FGETTOBJ TEXTOBJ TEXTLEN)))
             (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 CHNO)) ; Fileptrs are one back
             (SETQ CH (BIN TSTREAM)) ; The char at CHNO
             (CL:WHEN (AND (CHARCODEP CH)
                          (EQ PUNCT.TTC (\SYNCODE READSA CH)))
              ;; Started on a punct, return
              (RETURN CHNO))
             ;; Skip over any preceding whitespace characters. We don't know when the stream's piece might have changed, so test at each iteration.
             (for old CHNO from CHNO to 1 by -1 while (AND (CHARCODEP CH)
                                                            (EQ WHITESPACE.TTC (\SYNCODE READSA CH)))
              until (fetch CLPROTECTED of (PLOOKS (fetch (TEXTSTREAM PIECE) of TSTREAM)))
              do (SETQ CH (\BACKBIN TSTREAM)))
             ;;
             ;; CH is either TEXT, PUNCT, or image object.
             (CL:WHEN (AND (CHARCODEP CH)
                          (EQ PUNCT.TTC (\SYNCODE READSA CH)))
              ;; Punct before whitespace, look no further, punct is our guy.
              (RETURN (ADD1 CHNO)))
             ;;
             ;; We've reached the first unprotected non-separator character, and CHNO is the number of the character BEFORE that one. Continue
             ;; backwards through the text characters until reaching the first preceding non-text.
             (for old CHNO from CHNO to 1 by -1 until [OR (CL:IF (CHARCODEP CH)
                                                                (NEQ TEXT.TTC (\SYNCODE READSA CH))
                                                                T)
                 (fetch CLPROTECTED of (PLOOKS (fetch (TEXTSTREAM PIECE)
                                                       of TSTREAM))]
              do (SETQ CH (\BACKBIN TSTREAM)))
             (CL:WHEN (AND (CHARCODEP CH)
                          (EQ PUNCT.TTC (\SYNCODE READSA CH)))
              ;; We ended on a punct before some text, CHNO is one before the punct, get back to text
              (RETURN (IPLUS CHNO 2)))
             ;; We've now reached the first non-text character before the word, and CHNO is the character number of the character BEFORE it, or 0 if
             ;; you hit the front of the document. We add 1 for that, plus 1 to convert fileptr to charno.
             (RETURN (IPLUS CHNO (CL:IF (EQ CHNO 0)
                                       1
                                       2)))))
```

(\TEDIT.WORD.LAST

[LAMBDA (TSTREAM CHNO WORDBOUNDTABLE)

; Edited 29-Apr-2024 10:57 by rmk
; Edited 20-Mar-2024 10:54 by rmk
; Edited 17-Mar-2024 12:05 by rmk
; Edited 25-Dec-2023 18:38 by rmk
; Edited 23-May-2023 16:37 by rmk
; Edited 22-May-2023 10:52 by rmk
; Edited 29-May-91 18:22 by jds

;; Returns the number of the last character of the word containing CHNO or of the word following CHNO if CHNO does not map to a text character.
;; Unlike the paragraph case, we don't get much help from the pieces, because words are not piece-aligned. Caller can do the piece manipulation
;; given the result.

;; We don't need to worry about invisibles here,\BIN skips them.

;; Image objects are treated as text characters.

;; Punctuation is tricky: It stops whitespace and text, and its immediate successor doesn't matter.

```
(PROG* ((TEXTOBJ (TEXTOBJ! (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)))
       (READSA (fetch READSA of (OR WORDBOUNDTABLE (FGETTOBJ TEXTOBJ TXTWTBL)
                                                    TEDIT.WORDBOUND.READTABLE)))
       (TEXTLEN (FGETTOBJ TEXTOBJ TEXTLEN))
       (CH)
       (CL:WHEN (IGEQ CHNO TEXTLEN)
        (RETURN TEXTLEN))
       (SETQ CHNO (IMAX CHNO 1))
       (\TEDIT.TEXTSETFILEPTR TSTREAM (SUB1 CHNO)) ; Fileptrs are one back
       (SETQ CH (BIN TSTREAM)) ; The char at CHNO
       (CL:WHEN (AND (CHARCODEP CH)
                    (EQ PUNCT.TTC (\SYNCODE READSA CH)))
        ;; Started on a punct, return
        (RETURN CHNO)))
```

:: Skip over any following separator characters. Objects are consider to be text characters--don't skip over them. We don't know when the stream's piece might have changed, so test at each iteration.

```
(for old CHNO from CHNO to (SUB1 TEXTLEN) while (AND (CHARCODEP CH)
(EQ WHITESPACE.TTC (\SYNCODE READSA CH)))
until (fetch CLPROTECTED of (PLOOKS (fetch (TEXTSTREAM PIECE) of TSTREAM))
do (SETQ CH (BIN TSTREAM)))
```

:: CH is either TEXT, PUNCT, or image object.

```
(CL:WHEN (AND (CHARCODEP CH)
(EQ PUNCT.TTC (\SYNCODE READSA CH)))
```

:: Punct after whitespace, look no further, punct is our guy.

```
(RETURN CHNO))
```

:: We reached the last unprotected non-separator character, and CHNO is the number of the character AFTER that one. Continue forwards through the text characters until reaching the first following separator.

```
(for old CHNO from CHNO to (SUB1 TEXTLEN) until [OR (CL:IF (CHARCODEP CH)
(NEQ TEXT.TTC (\SYNCODE READSA CH))
T)
(fetch CLPROTECTED of (PLOOKS (fetch (TEXTSTREAM PIECE)
of TSTREAM])
```

```
do (SETQ CH (BIN TSTREAM)))
(CL:WHEN (AND (CHARCODEP CH)
(EQ PUNCT.TTC (\SYNCODE READSA CH)))
```

:: We ended on a punct after some text, CHNO is one after the punct, get back to text

```
(RETURN (SUB1 CHNO)))
```

:: We've now reached the first separator character after the word, and CHNO is the character number of the character after it, or TEXTLEN if we ran off the end..

```
(RETURN (CL:IF (IGEQ CHNO TEXTLEN)
TEXTLEN
(SUB1 CHNO)))
```

)

:: Would be nice to just do (DOFILESLOAD (CDR TEDITFILES)). But the order for exports.all and the order for loading have to be aligned.

```
(FILESLOAD TEDIT-FIND TEDIT-HISTORY TEDIT-FILE TEDIT-OLDFILE TEDIT-WINDOW TEDIT-SELECTION TEDIT-TFBRVAVO
TEDIT-HCPY TEDIT-PAGE TEDIT-BUTTONS TEDIT-MENU TEDIT-FNKEYS)
```

:: TEDIT Support information

```
(RPAQQ TEDITSYSTEMDATE "25-Feb-2025 15:07:03")
```

:: LISTFILES Interface, so the system can decide if a file is a TEdit file.

```
(ADDTOTVAR PRINTFILETYPES (TEDIT (TEST \TEDIT.GET.TRAILER)
(EXTENSION (TEDIT))))
```

FUNCTION INDEX

COERCETEXTOBJ	8	TEDIT.NTHCHARCODE	15	\TEDIT.FIRST.OBJPIECE	12
COERCETEXTSTREAM	4	TEDIT.OBJECT.CHANGED	11	\TEDIT.GET.PAGE.HEADINGS	12
EDIT-TEDIT	2	TEDIT.QUIT	7	\TEDIT.INSERT	16
MAKE-TEDIT-EXPORTS.ALL	2	TEDIT.RPLCHARCODE	14	\TEDIT.INSERT.SELPIECES	20
TDRIBBLE	9	TEDIT.STRINGWIDTH	7	\TEDIT.MOVE	17
TEDIT	3	TEDIT.TERPRI	7	\TEDIT.MOVE.MSG	13
TEDIT-SEE	6	TEDITMENUM	4	\TEDIT.NEXT.OBJPIECE	12
TEDIT.CHARWIDTH	8	TEDITSTRING	5	\TEDIT.PARA.FIRST	25
TEDIT.CONCAT	5	TEXTOBJ	8	\TEDIT.PARA.LAST	26
TEDIT.COPY	6	TEXTSTREAM	3	\TEDIT.PARAPIECES	25
TEDIT.DELETE	6	TEXTSTREAMP	4	\TEDIT.PIECE.NTHCHARCODE	15
TEDIT.EDIT.OBJECT	10	UPDATE-TEDIT	2	\TEDIT.READONLY	13
TEDIT.FIND.OBJECT	10	\TEDIT.APPLY.OBJFN	22	\TEDIT.REPLACE.SELPIECES	19
TEDIT.FIND.OBJECT.BACKWARD	10	\TEDIT.CHARDELETE	21	\TEDIT.RESTARTFN	20
TEDIT.INSERT	6	\TEDIT.CONCAT.INSTALL.HEADINGS	13	\TEDIT.THELP	25
TEDIT.INSERT.OBJECT	9	\TEDIT.CONCAT.PAGEFRAMES	12	\TEDIT.WORD.FIRST	26
TEDIT.KILL	7	\TEDIT.COPY	19	\TEDIT.WORD.LAST	27
TEDIT.MAP.OBJECTS	11	\TEDIT.COPYPIECE	21	\TEDIT.WORDDELETE	24
TEDIT.MOVE	7	\TEDIT.DELETE	22	\TEDIT.WORDDELETE.FORWARD	24
TEDIT.NCHARS	13	\TEDIT.DIFFUSE.PARALOOKS	23	\TEDIT1	16
TEDIT.NTHCHAR	15	\TEDIT.FINISHEDIT?	24		

VARIABLE INDEX

CHECK-TEDIT-ASSERTIONS	2	TEDIT.STARTUP.MONITORLOCK	3	TEDITSYSTEMDATE	28
PRINTFILETYPES	28	TEDIT.TENTATIVE	3	\TEDIT.THELPFLG	25
TEDIT.DEFAULT.PROPS	3	TEDITFILES	1		

MACRO INDEX

OBJECT.ALLOWS	2	TEDIT-ASSERT	2	TEVAL	8
---------------------	---	--------------------	---	-------------	---
