

File created: 1-Apr-2024 09:46:05 {WMEDLEY}<library>TEDIT>TEDIT.;538

edit by: rmk

previous date: 31-Mar-2024 10:14:22 {WMEDLEY}<library>TEDIT>TEDIT.;537

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **TEDITCOMS**

```
[ (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)
  [VARS (TEDIT.TERMSA.FONTS NIL)
    (TEDIT.TENTATIVE NIL)
    (TEDIT.DEFAULT.PROPS NIL)
    (TEDIT.STARTUP.MONITORLOCK (CREATE.MONITORLOCK 'TEDIT.STARTUP))
  (GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)
  ;; Unslashed functions. Public?
  (FNS TEDIT TEXTSTREAM TEXTSTREAMP TEDITMENUP COERCETEXTSTREAM TEDIT.CONCAT TEDITSTRING TEDIT-SEE
    TEDIT.COPY TEDIT.DELETE TEDIT.INSERT TEDIT.KILL TEDIT.QUIT TEDIT.MOVE TEDIT.STRINGWIDTH
    TEDIT.CHARWIDTH)
  (FNS TEXTOBJ COERCETEXTOBJ)
  (FNS \TEDIT.CONCAT.PAGEFRAMES \TEDIT.GET.PAGE.HEADINGS \TEDIT.CONCAT.INSTALL.HEADINGS
    \TEDIT.DO.BLUEPENDINGDELETE)
  (FNS \TEDIT.MOVE.MSG \TEDIT.READONLY)
  (FNS TEDIT.NCHARS TEDIT.RPLCHARCODE TEDIT.NTHCHARCODE \TEDIT.PIECE.NTHCHARCODE)
  ;; Slashed functions. Private?
  (FNS \TEDIT1 \TEDIT.INSERT \TEDIT.REPLACE.SELPIECES \TEDIT.INSERT.SELPIECES \TEDIT.OBJECT.SHOWSEL
    \TEDIT.RESTARTFN \TEDIT.CHARDELETE \TEDIT.CHARDELETE.FORWARD \TEDIT.COPYPIECE \TEDIT.APPLY.OBJFN
    \TEDIT.DELETE \TEDIT.DIFUSE.PARALOOKS \TEDIT.QUIT \TEDIT.WORDDELETE \TEDIT.WORDDELETE.FORWARD)
  (FNS \TEDIT.PARAPIECES \TEDIT.PARA.FIRST \TEDIT.PARA.LAST)
  (FNS \TEDIT.WORD.FIRST \TEDIT.WORD.LAST)
  (COMS ; Object-oriented editing
    (FNS TEDIT.INSERT.OBJECT TEDIT.EDIT.OBJECT TEDIT.FIND.OBJECT TEDIT.OBJECT.CHANGED
      TEDIT.MAP.OBJECTS)
    (FILES IMAGEOBJ))
  ;; 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-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))
```

;; Loadup stuff

```
(RPAQQ TEDITFILES (TEDIT TEDIT-PCTREE TEDIT-SELECTION TEDIT-SCREEN TEDIT-STREAM TEDIT-COMMAND TEDIT-FILE
  TEDIT-OLDFILE TEDIT-LOOKS TEDIT-WINDOW 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)

; Edited 26-Oct-2022 21:10 by rmk

:: updates sysout with new versions of loaded files. Keeps the extension

(FOR FILE DIRFILE LOADEDFILE INSIDE (OR FILES TEDITFILES)

WHEN [AND (SETQ LOADEDFILE (FOR F IN LOADEDFILELST WHEN (EQ FILE (FILENAMEFIELD F 'NAME)) DO (RETURN F)))

(SETQ DIRFILE (INFILEP (PACKFILENAME 'VERSION NIL 'BODY LOADEDFILE]

UNLESS (EQ LOADEDFILE DIRFILE) COLLECT (LOAD DIRFILE T)]

(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

(BKSYSBUF " ")

(RESETLST

(RESETSAVE LOADDBFLG '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)

[HELP "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)

(RPAQO TEDIT.TERMSA.FONTS NIL)

(RPAQO TEDIT.TENTATIVE NIL)

```
{MEDLEY}<library>tedit>TEDIT.;1
```

```
(RPAQQ TEDIT.DEFAULT.PROPS NIL)
```

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

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

```
(GLOBALVARS TEDIT.TENTATIVE TEDIT.DEFAULT.PROPS)
)
```

```
:: Unslashed functions. Public?
```

```
(DEFINEQ
```

## (TEDIT

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

```
;; 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)
```

```
; 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.
```

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

```
(SETQ WINDOW (\TEDIT.PRIMARYW TSTREAM))
```

```
(COND
```

```
(DONTSPAWN
```

```
(\TEDIT1 TSTREAM WINDOW T))
```

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

```
(T
```

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

```
; Spawn a process to do the edit.
```

```
'NAME
```

```
'Tedit
```

```
'RESTARTABLE
```

```
'HARDRESET
```

```
'RESTARTFORM
```

```
(LIST (FUNCTION \TEDIT.RESTARTFN)
```

```
(KWOTE TSTREAM)
```

```
WINDOW
```

```
(KWOTE PROPS]
```

```
(PROCESSPROP PROC 'WINDOW WINDOW)
```

```
(CL:UNLESS (LISTGET PROPS 'LEAVETTY)
```

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

```
(TTY.PROCESS PROC))
```

```
PROC])
```

## (TEXTSTREAM

```
[LAMBDA (TSTREAM? NOERROR)
```

```
; 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 (WINDOWP 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 (type? SELECTION TSTREAM?)
```

```
then (CL:WHEN (type? TEXTOBJ (SETQ X (FGETSEL TSTREAM? SELTEXTOBJ)))
```

```
(FGETTOBJ X STREAMHINT]
```

```
(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 TEXTOBJ 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 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 FMTSPEC (FGETTOBJ FIRSTTOBJ FMTSPEC))
    (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.UNIQIFY.ALL CTEXTOBJ))
  CSTREAM])
```

**(TEDITSTRING**

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

;;; 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)))
  (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 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.  
:: This results in a single history event.

```
(CL:UNLESS (\TEDIT.MOVE.MSG FROM TO T)
  (PROG ((TOBJ (GETSEL TO SELTEXTOBJ))
        (FOBJ (GETSEL FROM SELTEXTOBJ))
        FROMPIECES)
    (\TEDIT.SHOWSEL FROM NIL) ; Turn off any current highlighting
    (\TEDIT.SHOWSEL TO NIL))
```

:: Install FROM pieces at TO, first clearing out the blue pending delete.

::

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

```
(SETQ FROMPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES FROM)
    'COPY TOBJ FOBJ))
(CL:UNLESS FROMPIECES (RETURN))
```

::

:: Either replace or insert in the target, depending on its BLUEPENDINGDELETE

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

::

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

::

```
(\TEDIT.SHOWSEL TO NIL) ; Take down anything that might have appeared
(\TEDIT.FIXSEL TO TOBJ)
(\TEDIT.SHOWSEL TO T))]]
```

**(TEDIT.DELETE**

```
[LAMBDA (STREAM SEL LEN LEAVECARETLOOKS)
```

; 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 STREAM.

:: 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 STREAM)))
  (CL:WHEN (FIXP SEL)
    (TEDIT.SETSEL STREAM SEL LEN NIL NIL LEAVECARETLOOKS)
    (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)))
  (\TEDIT.DELETE TEXTOBJ (OR SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
```

**(TEDIT.INSERT**

```
[LAMBDA (TSTREAM TEXT CH#ORSEL LOOKS DONTSCROLL)
```

; 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

:: Insert TEXT (litatom or string) at the appropriate spot in the text.. No-op if given something else--should it error?

```
(CL:UNLESS (\TEDIT.READONLY TSTREAM)
  (PROG ((TEXTOBJ (TEXTOBJ TSTREAM)))
    (CL:WHEN (LITATOM TEXT)
      (SETQ TEXT (MKSTRING TEXT))))
```

:: Can't insert an empty string sensibly.

```
(CL:WHEN (OR (NOT (STRINGP TEXT))
             (ZEROP (NCHARS TEXT))))
  (RETURN))
```

```
(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)))
```

```
(CL:UNLESS (AND CH#ORSEL (GETSEL CH#ORSEL SET))
  (TEDIT.PROMPTPRINT TEXTOBJ "Please select a place for the insertion." T)
  (RETURN))
```

```
(CL:WHEN LOOKS ; Set up any specified looks.
```

```
(TEDIT.CARETLOOKS TSTREAM LOOKS))
(\TEDIT.INSERT TEXT CH#ORSEL TEXTOBJ DONTSCROLL))]]
```

**(TEDIT.KILL**

```
[LAMBDA (TSTREAM)
```

; 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 (CAR (FGETTOBJ TEXTOBJ \WINDOW)))
                    [PROCESSP (SETQ TEDPROC (WINDOWPROP TEDW 'PROCESS))
                      (NEQ TEDPROC (THIS.PROCESS))])
              (DEL.PROCESS TEDPROC)
              (TEDIT.DEACTIVATE.WINDOW TEDW) ]])
```

**(TEDIT.QUIT**

[LAMBDA (STREAM VALUE)

; 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 STREAM to terminate, and to return VALUE

```
(LET (MAINW (TEXTOBJ (TEXTOBJ STREAM)))
      (FSETTOBJ TEXTOBJ EDITFINISHEDFLG (OR VALUE T))
      (CL:WHEN [AND (FGETTOBJ TEXTOBJ \WINDOW)
                    (NEQ (SETQ MAINW (\TEDIT.PRIMARYW TEXTOBJ))
                        (PROCESSPROP (TTY.PROCESS)
                                     'WINDOW])
```

; tell the command loop to stop next time through

;; there is a main window of the stream, and it is not the window of the tty process, so give it the tty

```
(TTY.PROCESS (WINDOWPROP MAINW 'PROCESS))
(AND (NEQ (TTY.PROCESS)
          (THIS.PROCESS))
      (until [OR (NOT (WINDOWPROP MAINW 'PROCESS))
                 (PROCESS.FINISHEDP (WINDOWPROP MAINW 'PROCESS))
              do
```

; Wait until the Edit process has had a chance to go away before  
; continuing here.

(DISMISS) ] ]])

**(TEDIT.MOVE**

[LAMBDA (FROM TO)

; Edited 15-Mar-2024 13:54 by rmk  
; Edited 5-Mar-2024 00:22 by rmk  
; Edited 1-Mar-2024 20:13 by rmk  
; Edited 20-Feb-2024 20:07 by rmk  
; Edited 15-Feb-2024 21:27 by rmk  
; Edited 1-Feb-2024 23:30 by rmk  
; Edited 28-Jan-2024 22:27 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

;; 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 (\TEDIT.MOVE.MSG FROM TO NIL)
           (PROG (TOBJ (GETSEL TO SELTEXTOBJ))
                 (FOBJ (GETSEL FROM SELTEXTOBJ))
                 FROMPIECES)
           (\TEDIT.SHOWSEL FROM NIL)
           (\TEDIT.SHOWSEL TO NIL)
```

; Turn off any current highlighting

;; Install FROM pieces at TO, first clearing out the blue pending delete. The move-event may be a composite of both. If we are doing a move in the same textobject, the TO selection (= SEL for a CTRL-SHIFIT SELOPERATION) must be updated to reflect a preceding FROM-deletion.

;;  
;; Grab (a copy of) the source pieces, if image objects allow copying

```
(SETQ FROMPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES FROM)
                                         'COPY TOBJ FOBJ))
(CL:UNLESS FROMPIECES (RETURN))
```

;;  
;; Delete the FROM unless an object doesn't allow deletion.

```
(if (EQ TOBJ FOBJ)
    then ;; In this case, TOBJ is SEL and FOBJ is something else.
        (CL:UNLESS (\TEDIT.DELETE.SELPIECES FOBJ FROM)
                    (RETURN))
        (\TEDIT.UPDATE.LINES FOBJ 'DELETION FROM NIL NIL)
        (\TEDIT.SEL.DELETEDCHARS TO FROM) ; Adjust TO to after-deletion chnos.
    elseif (\TEDIT.DELETE FOBJ FROM)
    else (RETURN))
```

;;  
;; The deletion has been accomplished and lines have been adjusted accordingly.  
;; Either replace or insert in the target, depending on its BLUEPENDINGDELETE

```

(if (FGETTOBJ TOBJ BLUEPENDINGDELETE)
  then (FSETTOBJ TOBJ BLUEPENDINGDELETE NIL)
        (\TEDIT.REPLACE.SELPIECES FROMPIECES TOBJ TO)
  else (\TEDIT.INSERT.SELPIECES FROMPIECES TOBJ TO))
(\TEDIT.SET.SEL.LOOKS TO 'NORMAL)
(\TEDIT.FIXSEL TO TOBJ)
(\TEDIT.SHOWSEL TO T)
;; Create a :Move event that combines the insert with the prior delete/replace.If FOBJ=TOBJ, the delete and insert happened in the same
;; document. If the events happened in different documents, the delete event is already in FOBJ and we leave it there so that event can be
;; undone by a separate undo action in FOBJ. But we still include in the TOBJ move event: if we undo the move, we certainly want to undo
;; the insert. And if the delete event is still the next undoing candidate in FOBJ, it makes sense to do the FOBJ undo, otherwise not.
;; \TEDIT.UNDO.MOVE figures this out.

```

```

(LET [(INSEVENT (\TEDIT.POPEVENT TOBJ))
      (DELEVENT (CL:IF (EQ TOBJ FOBJ)
                        (\TEDIT.POPEVENT TOBJ)
                        (\TEDIT.LASTEVENT FOBJ)))]
  ;; We have to pop the INSEVENT so we can see the DELEVENT, then we push it back.
  ;; We coerce the insert event into a move. We save the entire DELEVENT so that \TEDIT.UNDO.MOVE can test to see
  ;; whether a foreign FOBJ is in the proper state.
  ;; If DELEVENT may be a :Replace, with THDELETEDPIECES.
  (SETH INSEVENT THACTION :Move)
  (SETH INSEVENT THOLDINFO (CONS DELEVENT (CL:IF (NEQ FOBJ TOBJ)
                                                  FOBJ)))
  (\TEDIT.HISTORYADD TOBJ INSEVENT))))

```

**(TEDIT.STRINGWIDTH**

```

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

```

**(TEDIT.CHARWIDTH**

```

[LAMBDA (CH FONT TERMSA)
  (* 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
    (* There IS a TERMTABLE to account for)
    (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)
              (SHOULDNT)))
    (T
     (* The usual case is to treat every character as a graphic.)
     (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])
```

)

(DEFINEQ

(\TEDIT.CONCAT.PAGEFRAMES

```
[LAMBDA (CTEXTOBJ TTEXTOBJECTS INITIALFILEPIECES)
```

```

; 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 FMTSPEC's 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 TTEXTOBJECTS 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 (fetch (FMTSPEC FMTPARATYPE) of PPARALOOKS))
  do (FSETPC PC PPARALOOKS (create FMTSPEC using PPARALOOKS FMTPARASUBTYPE _
                                                (CADR (ASSOC (fetch (FMTSPEC FMTPARASUBTYPE)
                                                                of PPARALOOKS)
                                                                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 TTEXTOBJECTS)
                                                                              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))])
```

**(\TEDIT.DO.BLUEPENDINGDELETE**

[LAMBDA (SEL TEXTOBJ) ; Edited 9-Mar-2024 11:33 by rmk  
; Edited 24-Dec-2023 00:01 by rmk  
; Edited 8-Jul-2023 22:48 by rmk  
; Edited 4-May-2023 00:05 by rmk  
; Edited 22-Apr-2023 18:31 by rmk  
; Edited 29-May-91 18:21 by jds

:: Check for blue-pending-delete, and do it if it's there.

:: Return T if the deletion was made. For people who need to know

```
(CL:WHEN (GETOBJ TEXTOBJ BLUEPENDINGDELETE)
  (\TEDIT.RESET.EXTEND.PENDING.DELETE SEL TEXTOBJ) ; Make it a normal selection again.
  (\TEDIT.DELETE TEXTOBJ SEL T))
```

(DEFINEQ

**(\TEDIT.MOVE.MSG**

[LAMBDA (FROM TO COPYFLG) ; 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 ((FOBJ (AND FROM (GETSEL FROM SET)
  (GETSEL FROM SELTEXTOBJ)))
  (TOBJ (AND TO (GETSEL TO SET)
  (GETSEL TO SELTEXTOBJ)))
  (TYPE (CL:IF COPYFLG
    "copy"
    "move")))
  (if (AND FOBJ TOBJ)
    then (if (EQ FOBJ TOBJ)
      then (\TEDIT.READONLY TOBJ)
      elseif (\TEDIT.READONLY TOBJ "Destination")
      else (AND (NOT COPYFLG)
        (\TEDIT.READONLY FOBJ "Source")))
    else (if FOBJ
      then (TEDIT.PROMPTPRINT TOBJ (CONCAT "Please select a destination for the " TYPE)
        T T)
      else (TEDIT.PROMPTPRINT FOBJ (CONCAT "Please select a source for the " TYPE)
        T T)
    T))
```

**(\TEDIT.READONLY**

[LAMBDA (TEXTOBJ TYPE) ; Edited 1-Feb-2024 17:33 by rmk  
; Edited 13-Nov-2023 11:26 by rmk

```
(SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
(CL:WHEN (FGETTOBJ TEXTOBJ TXTREADONLY)
  (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (OR TYPE "Text")
    " is read only--aborted"))
```

```

      T T)
    T]])
)

```

(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) ; 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.

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

```

(LET ((TEXTOBJ (TEXTOBJ TSTREAM))
      PC START-OF-PIECE)
  (DECLARE (SPECVARS START-OF-PIECE))
  (CL:WHEN (ILESSP N 0)
    (add N (ADD1 (FGETTOBJ TEXTOBJ TEXTLEN))))
  (CL:UNLESS (AND (IGEQL N 1)
    (ILEQL 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))
  (CL:WHEN (FGETTOBJ TEXTOBJ STREAMHINT)
    (replace (STREAM BINABLE) of (FGETTOBJ TEXTOBJ STREAMHINT) with NIL))

```

;; Chop off the suffix. Unless N was last in PC, the piece containing is new.

```

(\TEDIT.ALIGNEDPIECE (ADD1 N)
  TEXTOBJ)
(SETQ PC (\TEDIT.ALIGNEDPIECE N TEXTOBJ) ; Chop off the prefix. PC is now the singleton target
(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)))
    (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.UNIQUIFY.CHARLOOKS (CHARLOOKS.FROM.FONT NEWCHARLOOKS)
      TEXTOBJ)
      NEWCHARLOOKS)))
NEWCHARCODE])

```

(TEDIT.NTHCHARCODE

```

[LAMBDA (TSTREAM N) ; 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.

```

(LET ((TEXTOBJ START-OF-PIECE)
      (DECLARE (SPECVARS START-OF-PIECE))
      (if (type? SELECTION TSTREAM)
        then (SETQ TEXTOBJ (TEXTOBJ (FGETSEL TSTREAM SELTEXTOBJ)))
        (CL:UNLESS (EQL 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.PIECE.NTHCHARCODE  
[LAMBDA (TEXTOBJ PC OFFSET)

; 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.

```

(CL:WHEN (AND (IGEQ OFFSET 1)
              (ILEQ OFFSET (PLEN PC))))
  (LET ((PCONTENTS (PCONTENTS PC))
        (SELECTC (PTYPE PC)
                  (STRING.PTYPES
                    (NTHCHARCODE PCONTENTS OFFSET))
                  (THINFILE.PTYPE
                    (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                                    (SUB1 OFFSET)))
                    (BIN PCONTENTS))
                  (FATFILE1.PTYPE
                    (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                                    (SUB1 OFFSET)))
                    (create WORD
                            HIBYTE _ (PCHARSET PC)
                            LOBYTE _ (BIN PCONTENTS)))
                  (FATFILE2.PTYPE
                    (\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                                    (UNFOLD (SUB1 OFFSET)
                                                                2))))
                  (\WIN PCONTENTS))
        (UTF8.PTYPE [\SETFILEPTR PCONTENTS (IPLUS (PFPOS PC)
                                                    (ITIMES (SUB1 OFFSET)
                                                            (PBYTESPERCHAR PC)
                                                            (UTF8.INCCODEFN PCONTENTS))
                    (OBJECT.PTYPE PCONTENTS))
        (SUBSTREAM.PTYPE ; A substream stored as an object
          (HELP '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.
        (HELP '\TEDIT.PIECE.NTHCHARCODE]))])
)

```

;; Slashed functions. Private?

(DEFINEQ

(\TEDIT1

[LAMBDA (TSTREAM WINDOW UNSPAWNED)

; 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 TEXTOBJ) ; Run the editing engine
  (CLOSEW WINDOW) ; Close the edit window
  (\TEDIT.TEXTCLOSEF TSTREAM) ; Close the underlying files
  (replace (STREAM ACCESSBITS) of TSTREAM with BothBits) ; But leave the stream itself accessible
  (CL:WHEN (GETTEXTPROP TEXTOBJ 'AFTERQUITFN) ; Apply any post-window-close (and post-QUIT) function
            (APPLY* (GETTEXTPROP TEXTOBJ 'AFTERQUITFN)
                    WINDOW TSTREAM))
  (CL:WHEN UNSPAWNED ; We're not a distinct process: Send back the edited text in some
                    ; suitable form
    (COND

```

```

((NEQ T (FGETTOBJ TEXTOBJ EDITFINISHEDFLG))
 (PROG1 (FGETTOBJ TEXTOBJ EDITFINISHEDFLG)
        (FSETTOBJ TEXTOBJ EDITFINISHEDFLG NIL)))
((STRINGP (FGETTOBJ TEXTOBJ TXTFILE))
 (COERCETEXTOBJ TEXTOBJ 'STRINGP))
(T TSTREAM)))]

```

(\TEDIT.INSERT

```

[LAMBDA (INSERT SEL TEXTOBJ DONTSCROLL)
; Edited 17-Mar-2024 11:41 by rmk
; Edited 15-Mar-2024 13:32 by rmk
; Edited 9-Mar-2024 11:36 by rmk
; Edited 18-Feb-2024 15:30 by rmk
; Edited 9-Feb-2024 10:52 by rmk
; Edited 28-Jan-2024 23:29 by rmk
; Edited 14-Jan-2024 12:14 by rmk
; Edited 22-Dec-2023 22:03 by rmk
; Edited 20-Dec-2023 15:27 by rmk
; Edited 12-Nov-2023 12:28 by rmk
; Edited 19-Oct-2023 11:05 by rmk
; Edited 15-Oct-2023 16:01 by rmk
; Edited 18-Sep-2023 12:48 by rmk
; Edited 6-Jun-2023 15:23 by rmk
; Edited 2-Jun-2023 00:12 by rmk
; Edited 27-May-2023 10:45 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).

:: TEDIT.INSERT passes DONTSCROLL and asserts NOTINCREMENTAL, other calls do incremental scrolling.

:: SELECTION-SET test may be unnecessary here, TEDIT.INSERT already checks, not sure about the 2 other calls.

```

(CL:UNLESS (\TEDIT.READONLY TEXTOBJ)
 [if (NOT (AND SEL (GETSEL SEL SET)))
  then (TEDIT.PROMPTPRINT TEXTOBJ "Please select a place for the insertion." T)
  else ; Check for blue-pending-delete, and do it if it's there.
    (CL:WHEN (FGETTOBJ TEXTOBJ BLUEPENDINGDELETE)
      (\TEDIT.RESET.EXTEND.PENDING.DELETE SEL TEXTOBJ)
      ; Make it a normal selection again.
    )
  )

```

```

(\TEDIT.DELETE TEXTOBJ SEL)
(PROG ((CHNO (TEDIT.GETPOINT TEXTOBJ SEL))
      (PANES (FGETTOBJ TEXTOBJ \WINDOW))
      NCHARSADDED)
 (CL:WHEN PANES
  (CL:UNLESS DONTSCROLL (TEDIT.NORMALIZECARET TEXTOBJ SEL))
  (\TEDIT.SHOWSEL SEL NIL)) ; Turn off any old highlights

```

```

(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.
        ; 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.
    [\TEDIT.INSERTCH INSERT CHNO TEXTOBJ (FMEMB INSERT (CHARCODE (EOL CR LF)
      (SETQ NCHARSADDED 1)
    )
  )
  else ; It's maybe worth a scan here to avoid the heavier per-character complexity of \INSERTCH.
    (if [thereis CH instring INSERT suchthat (FMEMB CH (CHARCODE (EOL CR LF)
      then [for CH instring INSERT as NCH# from CHNO
        do (\TEDIT.INSERTCH CH NCH# TEXTOBJ (FMEMB CH (CHARCODE (EOL CR LF)
          else (\TEDIT.INSERTCH INSERT CHNO TEXTOBJ)
        (SETQ NCHARSADDED (NCHARS INSERT)))
      (FSETTOBJ TEXTOBJ \DIRTY T)
    )
  )

```

:: The model (piece table) is now correct: NCHARSADDED new characters have been added in front of CHNO.

```

(\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION CHNO NCHARSADDED (AND NIL DONTSCROLL))

```

:: The insertion happened before the original CHNO, which is now shifted back by NCHARSADDED. The next insertion should also come in front of that (now displaced) character.

```

(\TEDIT.UPDATE.SEL SEL (SUB1 (IPLUS CHNO NCHARSADDED))
 0
 'RIGHT)
(\TEDIT.SET.SEL.LOOKS SEL 'NORMAL)
(CL:UNLESS DONTSCROLL (TEDIT.NORMALIZECARET TEXTOBJ SEL))
(\TEDIT.SHOWSEL SEL T)
(for PANE in PANES do (\TEDIT.SET.WINDOW.EXTENT TEXTOBJ PANE]))

```

(\TEDIT.REPLACE.SELPIECES

```
[LAMBDA (SELPIECES TEXTOBJ SEL DONTDISPLAY) ; Edited 15-Mar-2024 13:32 by rmk
; Edited 17-Feb-2024 16:34 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 SELPIECES, either a string or SELPIECES. Produces a :Replace history event. TEXTOBJ will  
 ;; remember the inserted SELPIECES for undoing, our history event only has to keep track of the pieces it replaced. (and where they were).

```
(LET ((POINT (GETSEL SEL POINT))
      DELEVENT) ; Keep the SEL point in case
  (\TEDIT.SHOWSEL SEL NIL)
  (CL:WHEN (\TEDIT.DELETE TEXTOBJ SEL T)
    (SETQ DELEVENT (\TEDIT.LASTEVENT TEXTOBJ))
    (\TEDIT.INSERT.SELPIECES SELPIECES TEXTOBJ SEL DONTDISPLAY)
    ;; SELPIECES is now in the TEXTOBJ; we don't want the insert event
    (\TEDIT.POPEVENT TEXTOBJ)
    (SETH DELEVENT THACTION :Replace)
    (SETH DELEVENT THLEN (fetch (SELPIECES SPLEN) of SELPIECES))
    (SETH DELEVENT THPOINT POINT)
    (\TEDIT.UPDATE.SEL SEL NIL (fetch (SELPIECES SPLEN) of SELPIECES)
      POINT) ; Maintain the original point.
    (\TEDIT.SET.SEL.LOOKS SEL 'NORMAL)
    ;; Make sure SEL is off to guarantee turning on
    (\TEDIT.SHOWSEL SEL NIL))
  (\TEDIT.SHOWSEL SEL T))
```

### (\TEDIT.INSERT.SELPIECES

```
[LAMBDA (SELPIECES TEXTOBJ TARGETSEL DONTDISPLAY) ; 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 (AND SELPIECES (fetch (SELPIECES SPFIRST) of SELPIECES))
  (LET ((INSCH# (TEDIT.GETPOINT NIL 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)
    (\TEDIT.UPDATE.LINES TEXTOBJ 'INSERTION INSCH# SPLEN DONTDISPLAY)
    ;; Adjust SEL to select the inserted material, with point on the right.
    (\TEDIT.UPDATE.SEL (FGETTOBJ TEXTOBJ SEL)
      INSCH# SPLEN 'RIGHT NIL T)
    (\TEDIT.HISTORYADD TEXTOBJ (create TEDIT.HISTORYEVENT
      THACTION _ :Insert
      THLEN _ SPLEN
      THCH# _ INSCH#
      THFIRSTPIECE _ SPFIRST))))))
```

### (\TEDIT.OBJECT.SHOWSEL

```
[LAMBDA (TEXTOBJ SEL L1 ON PANE) ; Edited 24-Jan-2024 09:27 by rmk
; Edited 25-Nov-2023 15:48 by rmk
; Edited 14-Oct-2023 12:12 by rmk
; Edited 6-Jun-2023 15:28 by rmk
; Edited 1-May-2023 14:36 by rmk
; Edited 9-Apr-2023 15:37 by rmk
; Edited 12-Jun-90 17:50 by mitani
```

;; We are highlighting (or dehighlighting) a selection containing only a single image object if it appears in PANE

```
(PROG ((X0 (FGETSEL SEL X0))
      (Y (FGETLD L1 YBOT))
      (OBJ (FGETSEL SEL SELOBJ))
      (XOFFSET (DSPXOFFSET NIL PANE))
      (YOFFSET (DSPYOFFSET NIL PANE))
      (IMAGEFN (IMAGEOBJPROP (FGETSEL SEL SELOBJ)
        'WHENOPERATEDONFN))
      IMAGEBOX)
  (CL:UNLESS (AND IMAGEFN (INSIDE? (CREATEREGION 0 0 (WINDOWPROP PANE 'WIDTH)
    (WINDOWPROP PANE 'HEIGHT))
    X0 Y))
    (RETURN))
```

;; The selection is in the pane and has an image function

```
(SETQ IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
  (APPLY* (IMAGEOBJPROP OBJ 'IMAGEBOXFN)
    OBJ PANE)))
(CL:WHEN L1 ; If no line, why are we here??
  ;; We need to use the YBASE of the line rather than the YBOT, taking into account the object's descent.
  (SETQ Y (- (GETLD L1 YBASE)
    (fetch (IMAGEBOX YDESC) of IMAGEBOX)))
(RESETLST
  [RESETSAVE (DSPXOFFSET (IDIFFERENCE (IPLUS X0 XOFFSET)
    (fetch XKERN of IMAGEBOX))
    PANE)
  (LIST (FUNCTION DSPXOFFSET)
    XOFFSET
    (WINDOWPROP PANE 'DSP]
  (RESETSAVE (DSPYOFFSET (IPLUS Y YOFFSET)
    PANE)
  (LIST (FUNCTION DSPYOFFSET)
    YOFFSET PANE))
  (RESETSAVE (DSPCLIPPINGREGION (create REGION
    LEFT _ 0
    BOTTOM _ 0
    WIDTH _ (IMIN (GETSEL SEL DX)
      (IDIFFERENCE (FGETTOBJ TEXTOBJ WRIGHT)
        X0))
    HEIGHT _ (fetch YSIZE of IMAGEBOX))
    PANE)
  (LIST (FUNCTION DSPCLIPPINGREGION)
    (DSPCLIPPINGREGION NIL PANE)
    PANE))
  [AND IMAGEFN (ERSETQ (APPLY* IMAGEFN OBJ PANE (COND
    (ON 'HIGHLIGHTED)
    (T 'UNHIGHLIGHTED))
    SEL
    (FGETTOBJ TEXTOBJ STREAMHINT))])]
```

(\TEDIT.RESTARTFN

[LAMBDA (TSTREAM WINDOW PROPS)

; 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 full 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))) ; Unattach the window, so we don't do a redisplay.
  (SETTOBJ TEXTOBJ \WINDOW) ; Reopen, reattach
  (SETQ TSTREAM (OPENTEXTSTREAM TSTREAM WINDOW NIL NIL PROPS))
  (SETQ TEXTOBJ (fetch (TEXTSTREAM TEXTOBJ) of TSTREAM)) ; New stream maybe new textobj.
  (SETTOBJ TEXTOBJ \DIRTY ODIRTY) ; Preserve dirty
  (\TEDIT.COMMAND.LOOP TEXTOBJ) ; Run the editing engine
  (CLOSEW WINDOW) ; Close the edit window. WHY ??
  (\TEDIT.TEXTCLOSEF TSTREAM) ; Close the underlying files
  (replace (STREAM ACCESSBITS) of TSTREAM with BothBits) ; But leave the stream itself accessible
  ; Apply any post-window-close (and post-QUIT) function
  (CL:WHEN (GETTEXTPROP TEXTOBJ 'AFTERQUITFN)
    (APPLY* (GETTEXTPROP TEXTOBJ 'AFTERQUITFN)
      WINDOW TSTREAM))])
```

(\TEDIT.CHARDELETE

[LAMBDA (TEXTOBJ SEL)

; Edited 17-Mar-2024 00:27 by rmk
; Edited 23-Dec-2023 17:32 by rmk
; Edited 24-May-2023 22:50 by rmk
; Edited 22-May-2023 23:24 by rmk
; Edited 19-Apr-93 10:50 by jds

;; This creates a selection that specifies a deletion of the single character just before SEL's caret, and then uses that to delete the character. It
;; assumes that SEL can be smashed.

```
(SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
(PROG [FIRSTPIECE (CH# (SUB1 (TEDIT.GETPOINT TEXTOBJ SEL)
  (CL:WHEN (ILEQ CH# 0) ; Can't backspace past start of document
    (RETURN))
  ;; Back up to the first visible character--that's the target, unless it is protected
  (for PC START-OF-PIECE backpieces (SETQ FIRSTPIECE (\TEDIT.CHTOPC CH# TEXTOBJ T))
    declare (SPECVARS START-OF-PIECE) until (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC))
    do (if (VISIBLEPIECEP PC)
      then ;; Found the last character of the most recent visible piece
        (\TEDIT.UPDATE.SEL SEL (CL:IF (EQ PC FIRSTPIECE)
          CH#
          (SUB1 START-OF-PIECE))
          1
          'RIGHT)
```

```

(\TEDIT.DELETE TEXTOBJ SEL)
(RETURN)
elseif (NEQ PC FIRSTPIECE)
then (add START-OF-PIECE (IMINUS (PLEN PC)))

```

**(\TEDIT.CHARDELETE.FORWARD**

```

[LAMBDA (TEXTOBJ SEL) ; Edited 17-Mar-2024 00:27 by rmk
; Edited 24-Dec-2023 00:36 by rmk

```

:: This creates a selection specifying a deletion of the single character just after SEL's caret, and then uses that to delete the character. It assumes that SEL can be smashed.

```

(SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
(PROG (FIRSTPIECE (CH# (TEDIT.GETPOINT TEXTOBJ SEL)))
  (CL:WHEN (IGREATERP CH# (FGETTOBJ TEXTOBJ TEXTLEN)) ; Can't delete past the end
    (RETURN)))

```

:: Forward to the first visible character--that's the target, unless it is protected

```

(for PC START-OF-PIECE inpieces (SETQ FIRSTPIECE (\TEDIT.CHTOPC CH# TEXTOBJ T))
  declare (SPECVARS START-OF-PIECE) until (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC))
  do (if (VISIBLEPIECEP PC)
    then ; Delete the next visible character
      (\TEDIT.UPDATE.SEL SEL (CL:IF (EQ PC FIRSTPIECE)
        CH#
        (SUB1 START-OF-PIECE))
        1
        'RIGHT)
      (\TEDIT.DELETE TEXTOBJ SEL)
      (RETURN SEL)
    else (add START-OF-PIECE (PLEN PC)))

```

**(\TEDIT.COPYPIECE**

```

[LAMBDA (PC FROMOBJ TOOBJ UNPROTECT OPERATION PROMPTTEXTOBJ) ; 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))) ; No matter what, we need a fresh copy.
  (SETQ NEWPC (create PIECE using PC PNEW _ T PREVPIECE _ NIL NEXTPIECE _ NIL PTREENODE _ NIL))
  (SELECTC (PTYPE PC)
    (FILE.PTYPES (CL:WHEN CROSSCOPY

```

:: 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 SRCPCFILE (OPENSTREAM SRCPCFILE 'INPUT 'OLD ' ((TYPE TEXT))
[FSETPC PC PCONTENTS (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW
  \ ((:EXTERNAL-FORMAT , (STREAMPROP SRCPCFILE
    :EXTERNAL-FORMAT]
(COPYBYTES SRCPCFILE (PCONTENTS PC)
  (PFPOS PC)
  (IPLUS (PFPOS PC)
    (ITIMES (PLEN PC)
      (PBYTESPERCHAR PC)
    (FSETPC PC PFPOS 0)))

```

(STRING.PTYPES (change (PCONTENTS NEWPC) (CONCAT DATUM))) ; In case this is in the current insertion string

```

(OBJECT.PTYPE ; 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 CLSELHERE _ NIL)
    TOOBJ))
  elseif CROSSCOPY
  then (FSETPC NEWPC PLOOKS (\TEDIT.UNIQUIFY.CHARLOOKS (PLOOKS PC)
    TOOBJ))
    (FSETPC NEWPC PPARALOOKS (\TEDIT.UNIQUIFY.PARALOOKS (PPARALOOKS PC)
    TOBJ)))

```



(RETURN NEWPC))

(\TEDIT.APPLY.OBJFN

[LAMBDA (OBJ OPERATION FROMTOBJ TOTOBJ PROMPTTEXTTOBJ)

; 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 PROMPTTEXTTOBJ or FROMTOBJ, and return NIL. Otherwise, we return an object, either OBJ or a copy.

```
(CL:UNLESS TOTOBJ (SETQ TOTOBJ FROMTOBJ))
(PROG [NEWOBJ (OBJFN (IMAGEOBJPROP OBJ (SELECTQ OPERATION
(COPY 'COPYFN)
(MOVE 'WHENMOVEDFN)
(INSERT 'WHENINSERTEDFN)
(DELETE ; This may want to apply to the first pane?
'WHENDELETEDFN)
NIL]
(SETQ NEWOBJ (if (OBJFN
then (APPLY* OBJFN OBJ (fetch (TEXTOBJ STREAMHINT) of FROMTOBJ)
(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 PROMPTTEXTTOBJ FROMTOBJ)
(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.PRIMARYW TOTOBJ)
'DSP)
(fetch (TEXTOBJ STREAMHINT) of FROMTOBJ)
(fetch (TEXTOBJ STREAMHINT) of TOTOBJ))
'DON'T DONT]
(RETURN NIL))
(RETURN (OR (IMAGEOBJPROP NEWOBJ)
OBJ))
```

(\TEDIT.DELETE

[LAMBDA (TEXTOBJ TARGETSEL DONTDISPLAY)

; 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. 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 is correct unless DONTDISPLAY.

::

:: 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 (\TEDIT.READONLY TEXTOBJ)
(\DTEST TARGETSEL 'SELECTION)
(CL:WHEN (AND (FGETSEL TARGETSEL SET)
(IGEQ (FGETSEL TARGETSEL DCH)
1))
(LET ((SEL (FGETTOBJ TEXTOBJ SEL))
(TCH# (FGETSEL TARGETSEL CH#)))
(\TEDIT.SHOWSEL SEL NIL)
(CL:UNLESS DONTDISPLAY ; Make sure target is on-screen in the SELPANE
(AND NIL (TEDIT.NORMALIZECARET TEXTOBJ TARGETSEL)))
(CL:WHEN (\TEDIT.DELETE.SELPIECES TEXTOBJ TARGETSEL)
; Delete the selected characters (if objects allow)
```

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

```
(\TEDIT.UPDATE.LINES TEXTOBJ 'DELETION TARGETSEL NIL DONTDISPLAY)
(FSETTOBJ TEXTOBJ CARETLOOKS (\TEDIT.GET.INSERT.CHARLOOKS TEXTOBJ TARGETSEL))
```

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

```
(\TEDIT.SEL.DELETEDDCHS SEL TARGETSEL)
```

:: 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.

```
(\TEDIT.UPDATE.SEL SEL TCH# 0 'LEFT)
(\TEDIT.FIXSEL SEL TEXTOBJ)
(CL:UNLESS DONTDISPLAY (\TEDIT.SHOWSEL SEL T)
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 (PLOOKS _ (PPARALOOKS PRIORPC))
  inpieces
  (NEXTPIECE PRIORPC) until (PPARALAST PC) do
```

```
; Copy para looks info in from the left, up to the first para break.
```

```
(FSETPC PC PPARALOOKS PLOOKS)))
```

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

```
(CL:WHEN SUCCEEDINGPC
```

```
(for PC (PLOOKS _ (PPARALOOKS SUCCEEDINGPC))
  backpieces
  (PREVPIECE SUCCEEDINGPC) until (OR (EQ PC PRIORPC)
  (PPARALAST PC))
  do (FSETPC PC PPARALOOKS PLOOKS)))]])
```

(\TEDIT.QUIT

```
[LAMBDA (W NOFORCE)
```

```
; 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 by the default TEDIT.DEFAULT.MENUFN to perform the QUIT command.
```

```
(PROG* ((TEXTOBJ (TEXTOBJ! (fetch (TEXTWINDOW WTEXTOBJ) of W)))
  (QUITFNS (GETTEXTPROP TEXTOBJ 'QUITFN))
  QUITFLG RESP)
[for QUITFN inside QUITFNS until (OR (EQ QUITFLG 'DON'T)
  (EQ QUITFLG T))
```

```
do (SETQ QUITFLG (COND
  ((EQ QUITFN T))
  (QUITFN (APPLY* QUITFN W (FGETTOBJ TEXTOBJ STREAMHINT)
  TEXTOBJ
  (FGETTOBJ TEXTOBJ EDITPROPS]
```

```
(COND
```

```
(EQ QUITFLG 'DON'T)
```

```
:: The user supplied a QUITFN, and it returned 'DON'T', so just ignore all this Fooferaw and keep editing.
```

```
(RETURN))
```

```
[ (AND (FGETTOBJ TEXTOBJ \DIRTY)
  (NOT (FGETTOBJ TEXTOBJ MENUFLG))
  (NEQ QUITFNS T)
  (NEQ QUITFLG T))
```

```
; If this document has changed, check with the user to make sure
; he really wants to do it.
```

```
(FSETTOBJ TEXTOBJ EDITFINISHEDFLG (MOUSECONFIRM "Not saved yet; LEFT to Quit anyway." T
  (FGETTOBJ TEXTOBJ PROMPTWINDOW]
```

```
(T
```

```
; Go ahead and quit the next time we see the main command
; loop.
```

```
(FSETTOBJ TEXTOBJ EDITFINISHEDFLG T)))
```

```
(CL:WHEN [AND (FGETTOBJ TEXTOBJ \WINDOW)
  (NOT NOFORCE)
  (NEQ (\TEDIT.PRIMARYW TEXTOBJ)
  (PROCESSPROP (TTY.PROCESS)
  'WINDOW])
  (TTY.PROCESS (WINDOWPROP (\TEDIT.PRIMARYW TEXTOBJ)
  'PROCESS)))
  (RETURN (FGETTOBJ TEXTOBJ EDITFINISHEDFLG))
```

(\TEDIT.WORDDELETE

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

```
; 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
```

```
:: This deletes all characters from the character just before the caret to the beginning of the preceding word, skipping over separators to reach the
:: target word.
```

```
(TEXTOBJ! TEXTOBJ)
(CL:UNLESS SEL
 (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
(PROG ((LASTNO (SUB1 (TEDIT.GETPOINT SEL)))
       FIRSTNO))
```

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

```
(CL:WHEN (ILEQ LASTNO 0) ; Nothing to delete at start of file.
 (RETURN))
(SETQ FIRSTNO (\TEDIT.WORD.FIRST TEXTOBJ LASTNO))
(\TEDIT.UPDATE.SEL SEL FIRSTNO (ADD1 (IDIFFERENCE LASTNO FIRSTNO)))
(\TEDIT.DELETE TEXTOBJ SEL])
```

(\TEDIT.WORDDELETE.FORWARD

```
[LAMBDA (TEXTOBJ SEL) ; 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.

```
(TEXTOBJ! TEXTOBJ)
(CL:UNLESS SEL
 (SETQ SEL (FGETTOBJ TEXTOBJ SEL)))
(PROG ((FIRSTNO (TEDIT.GETPOINT SEL))
       LASTNO))
```

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

```
(CL:WHEN (IGREATERP FIRSTNO (FGETTOBJ TEXTOBJ TEXTLEN)) ; Nothing to delete at end of file.
 (RETURN))
(SETQ LASTNO (\TEDIT.WORD.LAST TEXTOBJ FIRSTNO))
(\TEDIT.UPDATE.SEL SEL FIRSTNO (ADD1 (IDIFFERENCE LASTNO FIRSTNO)))
(\TEDIT.DELETE TEXTOBJ SEL])
```

)

(DEFINEQ

(\TEDIT.PARAPIECES

```
[LAMBDA (SEL/FIRSTCHAR LASTCHAR TEXTOBJ) ; 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 TEXTOBJ (FGETSEL SEL/FIRSTCHAR SELTEXTOBJ))
        (SETQ FIRSTCHAR (FGETSEL SEL/FIRSTCHAR CH#))
        [SETQ LASTCHAR (CL:IF (EQ 0 (FGETSEL SEL/FIRSTCHAR DCH))
                              FIRSTCHAR
                              (SUB1 (FGETSEL SEL/FIRSTCHAR CHLIM))])
   elseif (type? TEDITHISTORYEVENT SEL/FIRSTCHAR)
     then (SETQ FIRSTCHAR (GETTH SEL/FIRSTCHAR THCH#))
          [SETQ LASTCHAR (SUB1 (IPLUS FIRSTCHAR (GETTH SEL/FIRSTCHAR THLEN))
                                (SETQ FIRSTCHAR SEL/FIRSTCHAR))
          (CL:UNLESS LASTCHAR (SETQ LASTCHAR FIRSTCHAR))]
   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)
  SPLEN _ (ADD1 (IDIFFERENCE (CAR LAST)
                              (CAR FIRST)))
  SPFIRSTCHAR _ (CAR FIRST)
  SPLASTCHAR _ (CAR LAST]))
```

(\TEDIT.PARA.FIRST

```
[LAMBDA (TEXTOBJ CHNO PROTECTEDNOTOK) ; 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 (fetch (SELPIECES SPFIRST) of CHNO))
```

```

      (SETQ START (fetch (SELPIECES SPFIRSTCHAR) of CHNO))
    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 (fetch (CHARLOOKS CLPROTECTED) of (PLOOKS PC)))
  do (add PLENTOT (PLEN PC)) finally ;; We overshot on PC, its NEXT is the winner. If no PC, we hit the text beginning
      (RETURN (CONS (IDIFFERENCE START PLENTOT)
                    (CL:IF PC
                      (NEXTPIECE PC)
                      (\TEDIT.FIRSTPIECE TEXTOBJ))))))

```

(\TEDIT.PARA.LAST

```

[LAMBDA (TEXTOBJ CHNO PROTECTEDNOTOK)
; 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 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 (fetch (SELPIECES SPLAST) of CHNO))
           [SETQ END (SUB1 (IDIFFERENCE (fetch (SELPIECES SPLASTCHAR) of CHNO)
                                         (PLEN CHPIECE))
                             (PLEN CHPIECE))]
      elseif (type? PIECE CHNO)
      then (SETQ CHPIECE CHNO)
           (SETQ END (\TEDIT.PCTOCH CHNO TEXTOBJ))
      else (SETQ CHPIECE (\TEDIT.CHTOPC (CL:IF (type? SELECTION CHNO)
                                               (SUB1 (FGETSEL CHNO CHLIM))
                                               CHNO)
                                         TEXTOBJ T))
           (SETQ END START-OF-PIECE)) ; Find the paragraph's last char
    ;; 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 (fetch (CHARLOOKS CLPROTECTED)
                                                                           of (PLOOKS PC)))
      finally (RETURN (CONS (IMIN (IPLUS END PLENTOT -1)
                                (FGETTOBJ TEXTOBJ TEXTLEN))
                            PC]))
  )

```

(DEFINEQ

(\TEDIT.WORD.FIRST

```

[LAMBDA (TEXTOBJ CHNO WORDBOUNDTABLE)
; 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.

```

(TEXTOBJ! TEXTOBJ)
(PROG ((TSTREAM (FGETTOBJ TEXTOBJ STREAMHINT))
  (READSA (fetch READSA of (OR WORDBOUNDTABLE (FGETTOBJ TEXTOBJ TXTWTBL)
                                TEDIT.WORDBOUND.READTABLE)))
  CH)
  (CL:WHEN (ILEQ CHNO 1) ; Beginning of document
    (RETURN 1))

```

```

    (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 (TEXTOBJ CHNO WORDBOUNDTABLE)

; 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.

```

(TEXTOBJ! TEXTOBJ)
(PROG ((TSTREAM (FGETTOBJ TEXTOBJ STREAMHINT))
      (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))))

)

:: Object-oriented editing

(DEFINEQ

**(TEDIT.INSERT.OBJECT**

[LAMBDA (OBJECT TSTREAM CH#)

; Edited 16-Mar-2024 00:08 by rmk  
; Edited 3-Mar-2024 13:01 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.

(CL:UNLESS (\TEDIT.READONLY 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.

(SETQ OBJPC (create PIECE  
PTYPE \_ OBJECT.PTYPE  
PCONTENTS \_ OBJECT  
PLEN \_ 1  
PLOOKS \_ (FGETTOBJ TEXTOBJ CARETLOOKS)))

(CL:WHEN (SETQ SUBSTREAM (IMAGEOBJPROP OBJECT 'SUBSTREAM))  
; The new piece we're inserting  
; If this is computed text in bulk, fix the length.

(FSETPC OBJPC PTYPE SUBSTREAM.PTYPE)  
(FSETPC OBJPC PLEN (TEXTLEN (fetch (TEXTSTREAM TEXTOBJ) of SUBSTREAM))))

(SETQ OBJSELPIECES  
(\TEDIT.SELPIECES.COPY (create SELPIECES  
SPLEN \_ 1  
SPFIRST \_ OBJPC  
SPLAST \_ OBJPC  
SPFIRSTCHAR \_ CH#  
SPLASTCHAR \_ CH#)

'INSERT TEXTOBJ))  
(CL:UNLESS OBJSELPIECES ; Copy may not be allowed  
(RETURN))

::

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

(SETQ SEL (TEXTSEL TEXTOBJ))  
(\TEDIT.SHOWSEL SEL NIL)  
(CL:WHEN (type? SELECTION CH#)  
(SETQ CH# (GETSEL CH# CH#)))  
(\TEDIT.DO.BLUEPENDINGDELETE SEL TEXTOBJ) ; Do the pending delete, if there is one.  
(CL:WHEN CH#

(\TEDIT.UPDATE.SEL SEL (IMIN CH# (ADD1 (TEXTLEN TEXTOBJ)))  
0  
'LEFT))

(\TEDIT.INSERT.SELPIECES OBJSELPIECES TEXTOBJ SEL)  
(TEDIT.NORMALIZECARET TEXTOBJ)  
(\TEDIT.SHOWSEL SEL T)))]

**(TEDIT.EDIT.OBJECT**

[LAMBDA (STREAM OBJ)

; 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

```
(LET ((TEXTOBJ (TEXTOBJ STREAM))
      SEL CH# SELOBJ EDITFN)
  [COND
    [(AND OBJ (IMAGEOBJP OBJ))
      (SETQ CH# (TEDIT.FIND.OBJECT TEXTOBJ OBJ))
      (COND
        (CH# (SETQ SEL (FGETTOBJ TEXTOBJ SCRATCHSEL))
              (\TEDIT.UPDATE.SEL SEL CH# 1)
              (SETSEL SEL SELOBJ OBJ)
              (SETSEL SEL SELTEXTOBJ TEXTOBJ)
              (\TEDIT.FIXSEL SEL TEXTOBJ))
        (T (TEDIT.PROMPTPRINT TEXTOBJ "Can't find specified object." T)
           (T (SETQ SEL (FGETTOBJ TEXTOBJ SEL))
              (SETQ OBJ (GETSEL SEL SELOBJ))
              (COND
                [OBJ ; OK There's an object selected. Edit it.
                  (SETQ EDITFN (IMAGEOBJPROP OBJ 'EDITFN))
                  (COND
                    ((AND EDITFN (APPLY* EDITFN OBJ)) ; If the editfn makes a change, update the screen.
                     (for LINE inside (GETSEL SEL L1) do (FSETLD LINE LDIRTY T))
                     (FSETTOBJ TEXTOBJ TXTNEEDSUPDATE T)
                     (\TEDIT.UPDATE.SCREEN TEXTOBJ))
                    (T ; No object selected.
                     (TEDIT.PROMPTPRINT TEXTOBJ "Please select an editable object first." T))
                    ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  ]
)
```

**(TEDIT.FIND.OBJECT**

```
[LAMBDA (TEXTOBJ OBJ) ; Edited 16-Mar-2024 10:03 by rmk
; Edited 6-Nov-2022 11:12 by rmk
; Edited 3-May-93 12:52 by jds
(for CH# from 1 by (PLEN PC) as PC inpieces (\TEDIT.FIRSTPIECE (TEXTOBJ TEXTOBJ))
 when (EQ OBJ (PCONTENTS PC)) do (RETURN CH#))
```

**(TEDIT.OBJECT.CHANGED**

```
[LAMBDA (TSTREAM OBJECT) ; 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
; Notify TEdit that an object has changed, and the display may
; need to be updated.
(LET ((TEXTOBJ (TEXTOBJ TSTREAM))
      OBJPIECE CHANGEDCH#)
  [SETQ OBJPIECE (find PC inpieces (\TEDIT.FIRSTPIECE TEXTOBJ) suchthat (EQ OBJECT (PCONTENTS PC))
                        ; Find the piece containing this object
                        (CL:UNLESS OBJPIECE (HELP "Changed OBJECT not found!?"))
                        (SETQ CHANGEDCH# (\TEDIT.PCTOCH OBJPIECE TEXTOBJ)) ; Get the CH# of the changed object
                        (\TEDIT.MARK.LINES.DIRTY TEXTOBJ CHANGEDCH# CHANGEDCH#) ; Mark affected lines
                        (replace (TEXTOBJ \DIRTY) of TEXTOBJ with T) ; And mark the document dirty.
                        (\TEDIT.SHOWSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
                          NIL)
                        (\TEDIT.UPDATE.SCREEN TEXTOBJ)
                        (\TEDIT.FIXSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
                          TEXTOBJ)
                        (\TEDIT.SHOWSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
                          T))
  ]
)
```

**(TEDIT.MAP.OBJECTS**

```
[LAMBDA (TEXTOBJ FN FNARG COLLECT?) ; 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 TEXTOBJ. 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)
(for CH# OBJ FNVAL from 1 by (PLEN PC) as PC inpieces (\TEDIT.FIRSTPIECE (TEXTOBJ TEXTOBJ))
 when (AND (EQ OBJECT.PTYPE (PTYPE PC))
           (type? IMAGEOBJ (SETQ OBJ (PCONTENTS PC)))
           (SETQ FNVAL (APPLY* FN CH# OBJ FNARG)))
 do (CL:WHEN COLLECT?
     (PUSH $$VAL (LIST CH# OBJ FNVAL)))
 (CL:WHEN (EQ FNVAL 'STOP)
 (GO $$OUT))
 finally (RETURN (DREVERSE $$VAL)))
)
```

{MEDLEY}<library>tedit>TEDIT.;1

Page 24

:: 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-TFBRAVO  
TEDIT-HCPY TEDIT-PAGE TEDIT-MENU TEDIT-FNKEYS)

:: TEDIT Support information

(RPAQQ **TEDITSYSTEMDATE** " 1-Apr-2024 09:46:05")

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

(ADDTOVAR **PRINTFILETYPES** (TEDIT (TEST \TEDIT.GET.TRAILER)  
(EXTENSION (TEDIT))))



---

**FUNCTION INDEX**

COERCETEXTOBJ .....	9	TEDIT.OBJECT.CHANGED .....	23	\TEDIT.GET.PAGE.HEADINGS .....	9
COERCETEXTSTREAM .....	4	TEDIT.QUIT .....	7	\TEDIT.INSERT .....	13
EDIT-TEDIT .....	2	TEDIT.RPLCHARCODE .....	11	\TEDIT.INSERT.SELPIECES .....	14
MAKE-TEDIT-EXPORTS.ALL .....	1	TEDIT.STRINGWIDTH .....	8	\TEDIT.MOVE.MSG .....	10
TEDIT .....	3	TEDITMENU .....	4	\TEDIT.OBJECT.SHOWSEL .....	14
TEDIT-SEE .....	5	TEDITSTRING .....	5	\TEDIT.PARA.FIRST .....	19
TEDIT.CHARWIDTH .....	8	TEXTOBJ .....	8	\TEDIT.PARA.LAST .....	20
TEDIT.CONCAT .....	4	TEXTSTREAM .....	3	\TEDIT.PARAPIECES .....	19
TEDIT.COPY .....	5	TEXTSTREAMP .....	4	\TEDIT.PIECE.NTHCHARCODE .....	12
TEDIT.DELETE .....	6	UPDATE-TEDIT .....	2	\TEDIT.QUIT .....	18
TEDIT.EDIT.OBJECT .....	22	\TEDIT.APPLY.OBJFN .....	17	\TEDIT.READONLY .....	10
TEDIT.FIND.OBJECT .....	23	\TEDIT.CHARDELETE .....	15	\TEDIT.REPLACE.SELPIECES .....	13
TEDIT.INSERT .....	6	\TEDIT.CHARDELETE.FORWARD .....	16	\TEDIT.RESTARTFN .....	15
TEDIT.INSERT.OBJECT .....	22	\TEDIT.CONCAT.INSTALL.HEADINGS ..	10	\TEDIT.WORD.FIRST .....	20
TEDIT.KILL .....	6	\TEDIT.CONCAT.PAGEFRAMES .....	9	\TEDIT.WORD.LAST .....	21
TEDIT.MAP.OBJECTS .....	23	\TEDIT.COPYPIECE .....	16	\TEDIT.WORDDELETE .....	18
TEDIT.MOVE .....	7	\TEDIT.DELETE .....	17	\TEDIT.WORDDELETE.FORWARD .....	19
TEDIT.NCHARS .....	11	\TEDIT.DIFFUSE.PARALOOKS .....	18	\TEDIT1 .....	12
TEDIT.NTHCHARCODE .....	11	\TEDIT.DO.BLUEPENDINGDELETE .....	10		

---

**VARIABLE INDEX**

CHECK-TEDIT-ASSERTIONS .....	2	TEDIT.STARTUP.MONITORLOCK .....	3	TEDITFILES .....	1
PRINTFILETYPES .....	24	TEDIT.TENTATIVE .....	2	TEDITSYSTEMDATE .....	24
TEDIT.DEFAULT.PROPS .....	3	TEDIT.TERMSA.FONT .....	2		

---

**MACRO INDEX**

OBJECT.ALLOWS .....	2	TEDIT-ASSERT .....	2
---------------------	---	--------------------	---

---