

File created: 20-Mar-2024 11:05:20 {WMEDLEY}<library>tedit>TEDIT-HISTORY.;154

edit by: rmk

changes to: (FNS TEDIT.UNDO)

previous date: 15-Mar-2024 13:55:42 {WMEDLEY}<library>tedit>TEDIT-HISTORY.;153

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

### (RPAQQ TEDIT-HISTORYCOMS

```
((DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS TEDITHISTORYEVENT)
                                         (MACROS \TEDIT.LASTEVENT \TEDIT.POPEVENT GETTH SETH)))
```

```
(FNS \TEDIT.HISTORYEVENT.DEFPRINT)
(INITRECORDS TEDITHISTORYEVENT)
(GLOBALVARS TEDIT.HISTORY.TYPELST TEDIT.HISTORYLST)
(INITVARS (TEDIT.HISTORY.TYPELST NIL)
          (TEDIT.HISTORYLST NIL))
```

```
(COMS ;; History-list maintenance functions
```

```
(FNS \TEDIT.HISTORYADD \TEDIT.CUMULATE.EVENTS)
```

```
(COMS ;; Specialized UNDO & REDO functions.
```

```
(FNS TEDIT.UNDO \TEDIT.UNDO1 TEDIT.REDO \TEDIT.UNDO.UNDO)
(FNS \TEDIT.UNDO.INSERTION \TEDIT.UNDO.DELETION \TEDIT.UNDO.MOVE \TEDIT.UNDO.REPLACE)
(FNS \TEDIT.REDO.INSERTION \TEDIT.REDO.REPLACE \TEDIT.REDO.MOVE)))
```

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

```
;; FOLLOWING DEFINITIONS EXPORTED
```

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

```
(DATATYPE TEDITHISTORYEVENT ( ;; Describes one event on the TEdit edit history list.
```

```
THACTION ; A keyword specifying what the event was
THPOINT ; Was the selection to the left or right?
THLEN ; The # of chars involved
THCH# ; The starting ch#
THFIRSTPIECE ; First piece involved
THOLDINFO ; Old info, for undo
NIL ; Was THAUXINFO: Auxiliary info about the event, primarily for
; redo
```

```
THDELETEDPIECES)
```

```
[ACCESSFNS TEDITHISTORYEVENT ((THCHLIM (AND (fetch (TEDITHISTORYEVENT THCH#) of DATUM)
                                             (IPLUS (fetch (TEDITHISTORYEVENT THCH#) of DATUM)
                                             (fetch (TEDITHISTORYEVENT THLEN) of DATUM]
```

```
(INIT (DEFPRINT 'TEDITHISTORYEVENT (FUNCTION \TEDIT.HISTORYEVENT.DEFPRINT)))
THPOINT _ 'LEFT)
```

```
)
```

```
(/DECLAREDATATYPE 'TEDITHISTORYEVENT ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
```

```
;; ---field descriptor list elided by lister---
```

```
' 16)
```

```
(DEFPRINT 'TEDITHISTORYEVENT (FUNCTION \TEDIT.HISTORYEVENT.DEFPRINT))
```

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

```
(PUTPROPS \TEDIT.LASTEVENT MACRO ((TOBJ)
                                   (CAR (fetch (TEXTOBJ TXTHISTORY) of TOBJ))))
```

```
(PUTPROPS \TEDIT.POPEVENT MACRO ((TOBJ)
                                   (POP (fetch (TEXTOBJ TXTHISTORY) of TOBJ))))
```

```
(PUTPROPS GETTH MACRO ((EVENT FIELD)
                       (fetch (TEDITHISTORYEVENT FIELD) of EVENT)))
```

```
(PUTPROPS SETTH MACRO ((EVENT FIELD NEWVALUE)
                       (replace (TEDITHISTORYEVENT FIELD) of EVENT with NEWVALUE)))
```

```
)
```

```
)
```

```
;; END EXPORTED DEFINITIONS
```

```
(DEFINEQ
```

```
(\TEDIT.HISTORYEVENT.DEFPRINT
```

```
[LAMBDA (EVENT STREAM)
```

```
; Edited 24-May-2023 23:36 by rmk
; Edited 22-May-2023 14:42 by rmk
; Edited 21-May-2023 09:15 by rmk
```

```
(LET (INFO LOC)
```

```

      (SETQ INFO (CONCAT (fetch (TEDITHISTORYEVENT THACTION) of EVENT)
        " "
        (fetch (TEDITHISTORYEVENT THCH#) of EVENT)
        "_ "
        (fetch (TEDITHISTORYEVENT THLEN) of EVENT)
        "_ "
        (NTHCHAR (fetch (TEDITHISTORYEVENT THPOINT) of EVENT)
          1)))
      (SETQ LOC (LOC EVENT))
      (CONS (CONCAT "{TH" ":" INFO " " (CAR LOC)
        "/"
        (CDR LOC)
        "}")
      )
)

```

```

(/DECLAREDATATYPE 'TEDITHISTORYEVENT ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  '16)

```

```

(DEFPRINT 'TEDITHISTORYEVENT (FUNCTION \TEDIT.HISTORYEVENT.DEFPRINT))
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS TEDIT.HISTORY.TYPELST TEDIT.HISTORYLST)
)

```

```

(RPAQ? TEDIT.HISTORY.TYPELST NIL)
(RPAQ? TEDIT.HISTORYLST NIL)

```

;; History-list maintenance functions

(DEFINEQ

(\TEDIT.HISTORYADD

```

[LAMBDA (TEXTOBJ EVENT)
  ; Edited 3-Mar-2024 12:15 by rmk
  ; Edited 19-Feb-2024 12:09 by rmk
  ; Edited 30-Dec-2023 22:19 by rmk
  ; Edited 11-Aug-2023 14:25 by rmk
  ; Edited 14-Jun-2023 16:04 by rmk
  ; Edited 12-Jun-2023 10:26 by rmk
  ; Edited 3-Jun-2023 20:41 by rmk
  ; Edited 28-May-2023 00:07 by rmk
  ; Edited 3-Sep-87 10:36 by jds

```

;; Add a new event to the history list, unless the list is currently DON'T (as in middle of foreign get).  
 ;; Not sure what should happen if the second one is to the right of the first, deleting forwards. Old code seemed to treat those as separate events,  
 ;; and only the second/right one could be undone.

```

(CL:UNLESS (EQ 'DON'T (GETTOBJ TEXTOBJ TXTHISTORY))
  (if (type? TEDITHISTORYEVENT EVENT)
    then (CL:WHEN (MEMB (GETTH EVENT THACTION)
      (CONSTANT (LIST :Put :Get))) ; Can't back up over Put/Get, flush the history.
      (FSETTOBJ TEXTOBJ TXTHISTORY NIL))
    )
)

```

;; Somebody may have already done there own fixup.

```

(LET ((OLDEVENT (\TEDIT.LASTEVENT TEXTOBJ))
  (CL:WHEN (AND (type? TEDITHISTORYEVENT OLDEVENT)
    (EQ :Delete (GETTH EVENT THACTION))
    (EQ :Delete (GETTH OLDEVENT THACTION)))
  )
)

```

;; Repeated successive deletions, we can combine them if they are adjacent.

```

(CL:WHEN (IEQP (GETTH EVENT THCHLIM)
  (GETTH OLDEVENT THCH#)) ; OLDEVENT is first, EVENT is still delete
  (SETQ EVENT (\TEDIT.CUMULATE.EVENTS EVENT OLDEVENT TEXTOBJ))
  (\TEDIT.POPEVENT TEXTOBJ) ; Pop OLDEVENT before repushing
  (SETQ OLDEVENT (\TEDIT.LASTEVENT TEXTOBJ)))
)

```

;; This may have created a new adjacency, if the accumulation of later deletes comes into with an earlier  
 ;; accumulation

```

(CL:WHEN [AND OLDEVENT (type? TEDITHISTORYEVENT OLDEVENT)
  (EQ :Delete (GETTH OLDEVENT THACTION))
  (IEQP (GETTH OLDEVENT THCHLIM)
    (IPLUS (GETTH EVENT THCH#)
      (GETTH OLDEVENT THLEN)
    )
  )
)

```

;; The OLDEEVENT deleted in front of EVENT, and itsTCHLIM are in its original coordinates. EVENT came  
 ;; later, with its TCH# in a coordinate system reduced by THLEN. So we have to add it back.

```

(SETQ EVENT (\TEDIT.CUMULATE.EVENTS OLDEVENT EVENT))
(\TEDIT.POPEVENT TEXTOBJ))
(push (GETTOBJ TEXTOBJ TXTHISTORY)
  EVENT)
)

```

elseif (LISTP EVENT)  
 then  
 ;; A monolithic sequence of undoable events

```
(push (GETTOBJ TEXTOBJ TXTHISTORY
      EVENT)))
```

(\TEDIT.CUMULATE.EVENTS

```
[LAMBDA (EVENT1 EVENT2 TEXTOBJ) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 3-Mar-2024 12:15 by rmk
; Edited 3-Jun-2023 17:09 by rmk
; Edited 27-May-2023 00:54 by rmk
; Edited 25-May-2023 23:58 by rmk
; Edited 21-May-2023 13:14 by rmk
; Edited 17-May-2023 14:55 by rmk
; Edited 3-Sep-87 10:42 by jds
```

:: Accumulate history events that should be combined into a undoable single even.  
 :: For now, this assumes they're events of the same type. Actually, this should be able to cumulate a delete/insert pair into a replacement, etc.

```
(SETH EVENT1 THDELETEDPIECES (\TEDIT.SELPIECES.CONCAT (GETTH EVENT1 THDELETEDPIECES)
              (GETTH EVENT2 THDELETEDPIECES)
              TEXTOBJ))
(SETH EVENT1 THLEN (fetch (SELPIECES SPLEN) of (GETTH EVENT1 THDELETEDPIECES))
EVENT1])
```

:: Specialized UNDO & REDO functions.

(DEFINEQ

(\TEDIT.UNDO

```
[LAMBDA (TEXTOBJ) ; Edited 20-Mar-2024 11:04 by rmk
; Edited 15-Mar-2024 13:36 by rmk
; Edited 7-Mar-2024 12:48 by rmk
; Edited 3-Mar-2024 20:02 by rmk
; Edited 22-Nov-2023 18:17 by rmk
; Edited 27-Sep-2023 00:14 by rmk
; Edited 23-Jun-2023 00:19 by rmk
; Edited 12-Jun-90 18:41 by mitani
```

:: Undo the last thing this guy did. This could be a sequence of subevents for a single user-level action that has more than one component: e.g.  
 :: move or replace is (Insert Delete). Undoing each (sub)event must restore the status quo ante (pieces, lines, looks, SEL).

:: We push information for undoing the undo onto the TXTHISTORYUNDO list.

```
(TEXTOBJ! TEXTOBJ)
(CL:UNLESS (FGETTOBJ TEXTOBJ TXTREADONLY)
; Only undo things if the document is allowed to change.
(TEDIT.PROMPTPRINT TEXTOBJ "" T)
(PROG ((SEL (TEXTSEL TEXTOBJ))
      (EVENT (\TEDIT.POPEVENT TEXTOBJ))
      PREVEVENTS UNDOEVENT)
      (CL:UNLESS EVENT
        (TEDIT.PROMPTPRINT TEXTOBJ "Nothing to undo" T)
        (RETURN))
```

:: Each main event was popped. Each subfunction must put back on the history-undo list one or more new events that would undo its  
 :: undoing.

:: We can get into trouble if there is an interrupt in the middle of undoing the full set of events for a previous action, or even in the middle of a  
 :: singleton event.

```
(SETQ PREVEVENTS (FGETTOBJ TEXTOBJ TXTHISTORY))
(\TEDIT.SHOWSEL SEL NIL)
(\TEDIT.UNDO1 TEXTOBJ EVENT)
```

:: Get the event that undid EVENT

```
(SETQ UNDOEVENT (\TEDIT.POPEVENT TEXTOBJ))
(FSETTOBJ TEXTOBJ TXTHISTORY PREVEVENTS)
(CL:WHEN [OR (NULL PREVEVENTS)
            (AND (type? TEDIT.HISTORYEVENT (CAR (LISTP PREVEVENTS)))
                (MEMB (GETTH (CAR PREVEVENTS)
                        THACTION)
                      (CONSTANT (LIST :Get :Put]
                                (SETTOBJ TEXTOBJ \DIRTY NIL))
```

:: The undone list keeps the event that would undo the undoing, the event that was just undone, and the history event that would be undone  
 :: next (by M-u). This is so that M-U can undo the undoing.

```
(push (FGETTOBJ TEXTOBJ TXTHISTORYUNDONE TEXTOBJ)
      (LIST (CAR PREVEVENTS)
            UNDOEVENT EVENT))
(\TEDIT.FIXSEL SEL TEXTOBJ)
(\TEDIT.SHOWSEL SEL T)))]
```

(\TEDIT.UNDO1

```
[LAMBDA (TEXTOBJ EVENT) ; Edited 4-Mar-2024 14:55 by rmk
; Edited 16-Jul-2023 11:14 by rmk
; Edited 30-May-2023 23:50 by rmk
```

; Edited 25-May-2023 00:33 by rmk

```
(SELECTC (GETH EVENT THACTION)
  ((LIST :Insert :Copy)
    (\TEDIT.UNDO.INSERTION TEXTOBJ EVENT))
  (:Move (\TEDIT.UNDO.MOVE TEXTOBJ EVENT))
  (:Delete (\TEDIT.UNDO.DELETION TEXTOBJ EVENT))
  (:Move (\TEDIT.UNDO.MOVE TEXTOBJ EVENT))
  (:Looks (\TEDIT.UNDO.LOOKS TEXTOBJ EVENT))
  (:ParaLooks (\TEDIT.UNDO.PARALOOKS TEXTOBJ EVENT))
  (:PageFormat [SETTOBJ TEXTOBJ TXTPAGEFRAMES (PROG1 (GETH EVENT THOLDINFO)
    (SETH EVENT THOLDINFO (GETTOBJ TEXTOBJ TXTPAGEFRAMES))
  ]))
)]
(\TEDIT.HISTORYADD TEXTOBJ EVENT)
((LIST :Replace :LowerCase :UpperCase) ;; He replaced one piece of text with another ; Lower-casing and upper-casing have the
  ;; same undo event.
  (\TEDIT.UNDO.REPLACE TEXTOBJ EVENT (GETH EVENT THACTION)))
(:Closefile (CL:WHEN (STREAMP (GETH EVENT THOLDINFO)
  (CLOSEF? (GETH EVENT THOLDINFO)))) ; Closes an included file
)
((LIST :Get :Put) ; He did a GET or PUT-- not undoable.
  (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "You can't undo a " (GETH EVENT THACTION)
  T))
(LET [(UNDOFN (CADDR (ASSOC (GETH EVENT THACTION)
  TEDIT.HISTORY.TYPELST)
  (COND
    (UNDOFN
      ;; DTEDIT.HISTORY.TYPELST is an ALST of form (type redofn undofn)
      (APPLY* UNDOFN TEXTOBJ EVENT (GETH EVENT THLEN)
        (GETH EVENT THCH#)
        (GETH EVENT THFIRSTPIECE)))
    (T (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "UNDO not implemented for " (GETH EVENT THACTION)
    T]))
  ]))

```

**TEDIT.REDO**

[LAMBDA (TEXTOBJ)

; Edited 15-Mar-2024 13:36 by rmk  
; Edited 4-Mar-2024 21:33 by rmk  
; Edited 2-Mar-2024 09:41 by rmk  
; Edited 21-Dec-2023 11:57 by rmk  
; Edited 27-May-2023 11:19 by rmk  
; Edited 30-May-91 21:27 by jds

;; REDO the last thing this guy did.

```
(CL:UNLESS (GETTOBJ TEXTOBJ TXTREADONLY)
  (PROG ((SEL (GETTOBJ TEXTOBJ SEL))
    (EVENT (\TEDIT.LASTEVENT TEXTOBJ))
    CH)
  (CL:UNLESS EVENT
    (TEDIT.PROMPTPRINT TEXTOBJ "Nothing to redo" T)
    (RETURN))
  (CL:UNLESS (GETSEL SEL SET)
    (TEDIT.PROMPTPRINT TEXTOBJ "Please select a target for the repeated action" T)
    (RETURN))
  )

```

;; There really is something to redo and something to do it to.

```
(\TEDIT.SHOWSEL SEL NIL)
(SELECTC (GETH EVENT THACTION)
  ((LIST :Insert :Copy :Move) ; It was an insertion
    (\TEDIT.REDO.INSERTION TEXTOBJ EVENT SEL))
  (:Delete (\TEDIT.DELETE TEXTOBJ SEL)) ; It was a deletion
  (:Replace (\TEDIT.REDO.REPLACE TEXTOBJ EVENT (GETH EVENT THACTION))) ; It was a replacement (a del/insert combo)
  (:LowerCase (\TEDIT.LCASE.SEL TEXTOBJ TEXTOBJ SEL)) ; He lower-cased something
  (:UpperCase (\TEDIT.UCASE.SEL TEXTOBJ TEXTOBJ SEL)) ; He upper-cased something
  (:Looks (TEDIT.LOOKS TEXTOBJ (PLOOKS (GETH EVENT THFIRSTPIECE)
    SEL)) ; It was a character looks change
  (:ParaLooks (TEDIT.PARALOOKS TEXTOBJ (PPARALOOKS (GETH EVENT THFIRSTPIECE)
    SEL)) ; It was a Paragraph looks change
  (:PageFormat (TEDIT.PROMPTPRINT TEXTOBJ "You can't redo a page-format change" T T))
  (:Find ; EXACT-MATCH SEARCH COMMAND
    (* (* ;; "RESTLST ?") (AND NIL (RESETSAVE
      (CURSOR WAITINGCURSOR)))
      (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)
      (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
      (\TEDIT.SHOWSEL SEL NIL) (SETQ CH
      (TEDIT.FIND TEXTOBJ (GETH EVENT THAUXINFO)))
      (COND (CH (TEDIT.PROMPTPRINT TEXTOBJ "done."))
    )
  )
  )

```

```

)
(\TEDIT.UPDATE.SEL SEL CH (NCHARS
(GETTH EVENT THAUXINFO)) (QUOTE RIGHT))
(\TEDIT.FIXSEL SEL TEXTOBJ)
(\TEDIT.NORMALIZECARET TEXTOBJ)
(\TEDIT.SHOWSEL SEL T)) (T (TEDIT.PROMPTPRINT
TEXTOBJ "[Not found]"))))
)
(:Move ; He moved some text
(\TEDIT.REDO.MOVE TEXTOBJ EVENT (GETTH EVENT THLEN)
(IMAX 1 (TEDIT.GETPOINT NIL SEL))
(GETTH EVENT THFIRSTPIECE)))
((LIST :Get :Put) ; Why can't you redo a get or put ?
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "You can't redo a " (GETTH EVENT THACTION))
T T))
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "Redoing the action " (GETTH EVENT THACTION)
" isn't implemented."))
T))
(\TEDIT.SHOWSEL SEL T)))]])

```

(\TEDIT.UNDO.UNDO

```

[LAMBDA (TEXTOBJ) ; Edited 3-Mar-2024 21:27 by rmk
; Edited 13-Jun-2023 15:05 by rmk
; Edited 3-Jun-2023 23:04 by rmk
; Edited 1-Jun-2023 23:53 by rmk

```

;; This undoes a preceding undo, by pushing the undoing event on the history list, and undoing that.  
;; The state is recorded as the event that would be undone next  
;; This makes sense only if the document is now in the state immediately after the undoing--if any other events have intervened, the character  
;; positions and the general state of the document are unrelated. So the elements of the undo list also contain the state of the (forward) history list  
;; after the undoing was undone. If we have moved back to the same point in history, we can do the undoing.

```

(SETQ TEXTOBJ (TEXTOBJ TEXTOBJ))
(TEDIT.PROMPTPRINT TEXTOBJ "" T)
(LET [(LASTUNDONE (pop (FGETTOBJ TEXTOBJ TXTHISTORYUNDONE))
(if (NULL LASTUNDONE)
then (TEDIT.PROMPTPRINT TEXTOBJ "There is no action whose undoing can be reversed" T)
elseif (EQ (CAR LASTUNDONE)
(\TEDIT.LASTEVENT TEXTOBJ))
then ;; We tell TEDIT.UNDO that LASTUNDONE is the one we now want to undo.
(push (FGETTOBJ TEXTOBJ TXTHISTORY)
(CADR LASTUNDONE))
(TEDIT.UNDO TEXTOBJ)
;; This saved what we just undid, don't want to keep reundoing it.
(pop (FGETTOBJ TEXTOBJ TXTHISTORYUNDONE))
(push (FGETTOBJ TEXTOBJ TXTHISTORY)
(CADDR LASTUNDONE))
else (SETTOBJ TEXTOBJ TXTHISTORYUNDONE NIL) ; If something else has happened, there are no undos to undo.
(TEDIT.PROMPTPRINT TEXTOBJ "Cannot undo the previous undo" T))
)
)

```

(DEFINEQ

(\TEDIT.UNDO.INSERTION

```

[LAMBDA (TEXTOBJ EVENT) ; Edited 30-May-2023 22:54 by rmk
; Edited 26-May-2023 23:49 by rmk
; Edited 24-May-2023 23:53 by rmk
; Edited 2-May-2023 23:26 by rmk
; Edited 21-Apr-93 01:33 by jds

```

;; UNDO a prior Insert, Copy, or Include.  
(\TEDIT.DELETE TEXTOBJ (\TEDIT.UPDATE.SEL (fetch (TEXTOBJ SEL) of TEXTOBJ
EVENT])

(\TEDIT.UNDO.DELETION

```

[LAMBDA (TEXTOBJ EVENT) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 30-May-2023 23:31 by rmk
; Edited 27-May-2023 23:39 by rmk
; Edited 21-Apr-93 12:01 by jds

```

;; UNDO a prior deletion  
(\TEDIT.INSERT.SELPIECES (\TEDIT.SELPIECES.COPY (GETTH EVENT THDELETEDPIECES)
'INSERT TEXTOBJ)
TEXTOBJ
(GETTH EVENT THCH#])

(\TEDIT.UNDO.MOVE

```

[LAMBDA (TEXTOBJ EVENT) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 4-Mar-2024 16:08 by rmk

```

;; If the deletion from TEDIT.MOVE was not in TEXTOBJ, the FOBJ must have been a separate document. If FOBJ is still in the state just after that  
;; deletion, it can be undone there. But if FOBJ is not in that state, undoing doesn't there make sense. The deleted string would reappear in some  
;; random place.

```
(LET ((DELEVENT (CAR (GETTH EVENT THOLDINFO)))
      (FOBJ (CDR (GETTH EVENT THOLDINFO)))
      (SEL (FGETTOBJ TEXTOBJ SEL)))
      (\TEDIT.DELETE TEXTOBJ (\TEDIT.UPDATE.SEL SEL EVENT)) ; Undo the insert in this document
      (CL:WHEN (GETTH EVENT THDELETEDPIECES) ;:Move must have started as :Replace
        (\TEDIT.INSERT.SELPIECES (\TEDIT.SELPIECES.COPY (GETTH EVENT THDELETEDPIECES)
          'INSERT TEXTOBJ)
          TEXTOBJ
          (GETTH EVENT THCH#)))
      (if FOBJ
        then (CL:WHEN (EQ DELEVENT (\TEDIT.LASTEVENT FOBJ)) ; Delete is last event in other document
          (\TEDIT.UNDO FOBJ))
        else (\TEDIT.UNDO1 TEXTOBJ DELEVENT))
      ;; Put the point back after the original target. Caller wil fix it.
      (\TEDIT.UPDATE.SEL SEL EVENT 0 'LEFT T]))
```

**(\TEDIT.UNDO.REPLACE**

```
[LAMBDA (TEXTOBJ EVENT ACTION) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 30-May-2023 23:10 by rmk
; Edited 27-May-2023 16:49 by rmk
; Edited 24-May-2023 22:43 by rmk
```

;; This undoes the replacement, but tracks for REDO whether the action was replace, lowercase, or uppercase.

```
(\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY (GETTH EVENT THDELETEDPIECES)
  NIL TEXTOBJ)
  TEXTOBJ
  (\TEDIT.UPDATE.SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
    EVENT))
(SETTH (\TEDIT.LASTEVENT TEXTOBJ)
  THACTION ACTION])
```

)

(DEFINEQ

**(\TEDIT.REDO.INSERTION**

```
[LAMBDA (TEXTOBJ EVENT SEL) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 31-May-2023 10:26 by rmk
; Edited 18-May-2023 19:24 by rmk
; Edited 21-Apr-93 01:06 by jds
```

;; Copies of the pieces inserted at the previous insertion EVENT are inserted at SEL's caret. We can extract the relevant pieces from the event's text position, because we know that either EVENT was the last event or other events after it have been undone, and the pieces are back to their original state.

```
(\TEDIT.INSERT.SELPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES EVENT NIL TEXTOBJ)
  'INSERT TEXTOBJ)
  TEXTOBJ SEL])
```

**(\TEDIT.REDO.REPLACE**

```
[LAMBDA (TEXTOBJ EVENT ACTION) ; Edited 15-Mar-2024 13:54 by rmk
; Edited 2-Oct-2023 11:43 by rmk
; Edited 31-May-2023 10:25 by rmk
; Edited 27-May-2023 11:16 by rmk
; Edited 16-May-2023 22:05 by rmk
; Edited 30-May-91 21:28 by jds
```

;; We get the replacement from where EVENT just installed it in the text (assume that it is still there unchanged), and then we use it to replace what is now at the current selection. EVENT's deleted pieces are not relevant.

```
(\TEDIT.REPLACE.SELPIECES (\TEDIT.SELPIECES.COPY (\TEDIT.SELPIECES EVENT NIL TEXTOBJ)
  NIL TEXTOBJ)
  TEXTOBJ
  (\TEDIT.UPDATE.SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
    EVENT))
(SETTH (\TEDIT.LASTEVENT TEXTOBJ)
  THACTION ACTION])
```

**(\TEDIT.REDO.MOVE**

```
[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 15-Mar-2024 13:36 by rmk
; Edited 16-Feb-2024 23:36 by rmk
; Edited 7-Jun-2023 23:19 by rmk
; Edited 27-May-2023 11:18 by rmk
; Edited 23-May-2023 12:54 by rmk
; Edited 30-May-91 21:28 by jds
```

```
(LET ((SCR2 (GETTOBJ TEXTOBJ SCRATCHSEL2)))
  (\TEDIT.UPDATE.SEL SCR2 (GETTH EVENT THCH#)
    LEN)
  (SETSEL SCR2 SET T)
  (\TEDIT.FIXSEL SCR2 TEXTOBJ)
  (\TEDIT.SET.SEL.LOOKS SCR2 'MOVE)
  (TEDIT.MOVE SCR2 (FGETTOBJ TEXTOBJ SEL]))
```

)



---

**FUNCTION INDEX**

TEDIT.REDO .....	4	\TEDIT.REDO.INSERTION .....	6	\TEDIT.UNDO.MOVE .....	5
TEDIT.UNDO .....	3	\TEDIT.REDO.MOVE .....	6	\TEDIT.UNDO.REPLACE .....	6
\TEDIT.CUMULATE.EVENTS .....	3	\TEDIT.REDO.REPLACE .....	6	\TEDIT.UNDO.UNDO .....	5
\TEDIT.HISTORYADD .....	2	\TEDIT.UNDO.DELETION .....	5	\TEDIT.UNDO1 .....	3
\TEDIT.HISTORYEVENT.DEFPRINT .....	1	\TEDIT.UNDO.INSERTION .....	5		

---

**MACRO INDEX**

GETH .....	1	SETH .....	1	\TEDIT.LASTEVENT .....	1	\TEDIT.POPEVENT .....	1
------------	---	------------	---	------------------------	---	-----------------------	---

---

**VARIABLE INDEX**

TEDIT.HISTORY.TYPELST ...	2	TEDIT.HISTORYLST .....	2
---------------------------	---	------------------------	---

---

**RECORD INDEX**

TEDITHISTORYEVENT .....	1
-------------------------	---

---