

File created: 4-May-95 10:38:22 {DSK}<lispcore>library>new>TEDITHISTORY.;3

changes to: (FNS TEDIT.REDO.INSERTION \TEDIT.CUMULATE.EVENTS TEDIT.UNDO TEDIT.UNDO.REPLACE)

previous date: 22-Mar-95 18:20:17 {DSK}<lispcore>library>new>TEDITHISTORY.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1983, 1984, 1985, 1986, 1987, 1990, 1991, 1993, 1995 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ TEDITHISTORYCOMS
  ((FILES TEDITDECLS)
   (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (\SCRATCHLEN 64))
    (FILES (LOADCOMP)
            TEDITDECLS))
   (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.UNDO.INSERTION TEDIT.UNDO.DELETION TEDIT.REDO TEDIT.REDO.INSERTION
                TEDIT.UNDO.MOVE TEDIT.UNDO.REPLACE TEDIT.REDO.REPLACE TEDIT.REDO.MOVE)))

(FILESLoad TEDITDECLS)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ \SCRATCHLEN 64)

(CONSTANTS (\SCRATCHLEN 64)
)

(FILESLoad (LOADCOMP)
            TEDITDECLS)
)

(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-Sep-87 10:36 by jds
   ;; Add a new event to the history list. For now, this just re-sets the whole list to be the one event..
   ;; This function also takes care of cumulating cumulative events, like successive deletions.
   (LET* ((OLDEVENT (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ))
           (ETYPE (fetch (TEDITHISTORYEVENT THACTION) of EVENT))
           (OETYPE (fetch (TEDITHISTORYEVENT THACTION) of OLDEVENT))
           (REALEVENT EVENT))
     [COND
      ((AND OLDEVENT (EQ OETYPE ETYPE))
       (EQ ETYPE 'Delete)) ; Repeated successive deletions. See if we can combine them.
      (LET* [(OSTART (fetch (TEDITHISTORYEVENT THCH#) of OLDEVENT))
              (NSTART (fetch (TEDITHISTORYEVENT THCH#) of EVENT))
              (OLDEND (+ OSTART (fetch (TEDITHISTORYEVENT THLEN) of OLDEVENT)))
              (NEWEND (+ NSTART (fetch (TEDITHISTORYEVENT THLEN) of EVENT)))]
        [COND
         ((IEQP OLDEND NSTART) ; The old deletion was just in front of the current one; cumulate
          ; them.
          (SETQ REALEVENT (\TEDIT.CUMULATE.EVENTS OLDEVENT EVENT T)))
         ((IEQP NEWEND OSTART) ; The new deletion was just in front of the old one; cumulate
          ; them.
          (SETQ REALEVENT (\TEDIT.CUMULATE.EVENTS EVENT OLDEVENT T))]
        (replace (TEXTOBJ TXTHISTORY) of TEXTOBJ with REALEVENT]))
```

(**TEDIT.CUMULATE.EVENTS**

[LAMBDA (EVENT1 EVENT2 PIECES-TO-SAVE?) ; Edited 3-Apr-95 12:23 by sybalsky:mv:envos

:: Accumulate history events that should really be combined into a single event.  
:: 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.

```
(LET* [(OLDLEN (fetch (TEDITHISTORYEVENT THLEN) of EVENT1))
      (NEWPC1 (CAR (fetch (TEDITHISTORYEVENT THFIRSTPIECE) of EVENT2)))
      (REALEVENT (create TEDITHISTORYEVENT using EVENT1 THLEN _ (+ OLDLEN (fetch (TEDITHISTORYEVENT THLEN) of EVENT2))
      (bind (PC _ (CAR (fetch (TEDITHISTORYEVENT THFIRSTPIECE) of EVENT1)))
            (CHCOUNT _ 0) while (< (SETQ CHCOUNT (+ CHCOUNT (fetch (PIECE PLEN) of PC)))
                                OLDLEN)
            do (SETQ PC (fetch (PIECE NEXTPIECE) of PC)) finally (replace (PIECE NEXTPIECE) of PC with NEWPC1)
                (replace (PIECE PREVPIECE) of NEWPC1 with PC)
                (RETURN))
      REALEVENT])
)
```

:: Specialized UNDO & REDO functions.

(DEFINEQ

(**TEDIT.UNDO**

[LAMBDA (TEXTOBJ) ; Edited 22-Mar-95 16:48 by sybalsky:mv:envos

:: Undo the last thing this guy did.

```
(COND
  ((NOT (fetch (TEXTOBJ TXTREADONLY) of TEXTOBJ))
   ;; Only undo things if the document is allowed to change.
   (PROG ((SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
          (EVENT CH# LEN FIRSTPIECE))
     (COND
      ((SETQ EVENT (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ))
       ; There really is something to UNDO. Decide what, & fix it.
       (SETQ LEN (fetch THLEN of EVENT)) ; Length of the text that was inserted/deleted/changed
       (SETQ CH# (fetch THCH# of EVENT)) ; Starting CH# of the change
       (SETQ FIRSTPIECE (CAR (fetch THFIRSTPIECE of EVENT))) ; First piece affected by the change
       (RESETLST
        (RESETSAVE (CURSOR WAITINGCURSOR))
        (\SHOWSEL SEL NIL NIL)
        [SELECTQ (fetch THACTION of EVENT)
          ((Insert Copy Include) ; It was an insertion
           (TEDIT.UNDO.INSERTION TEXTOBJ EVENT LEN CH# FIRSTPIECE))
          (Delete) ; It was a deletion
           (TEDIT.UNDO.DELETION TEXTOBJ EVENT LEN CH# FIRSTPIECE))
          (Looks) ; It was a character-looks change
           (TEDIT.UNDO.LOOKS TEXTOBJ EVENT LEN CH# FIRSTPIECE))
          (ParaLooks) ; It was a PARA looks change
           (TEDIT.UNDO.PARALOOKS TEXTOBJ EVENT LEN CH# FIRSTPIECE))
          (Move (TEDIT.UNDO.MOVE TEXTOBJ EVENT LEN CH# FIRSTPIECE)) ; He moved some text
        ]
       ((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 LEN CH# FIRSTPIECE))
        (Get) ; He did a GET -- not undoable.
        (TEDIT.PROMPTPRINT TEXTOBJ "You can't UNDO a GET." T))
        (Put) ; He did a PUT -- not undoable.
        (TEDIT.PROMPTPRINT TEXTOBJ "You can't UNDO a Put." T))
      (COND
       ((AND (SETQ UNDOFN (ASSOC (fetch THACTION of EVENT)
                                TEDITHISTORY.TYPELST))
             (SETQ UNDOFN (CADDR UNDOFN)))
        ; TEDIT.HISTORY.TYPELST is an ALST of form (type redofn
        ; undofn)
        (APPLY* UNDOFN TEXTOBJ EVENT LEN CH# FIRSTPIECE))
       (T (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "UNDO not implemented for "
                                             (fetch THACTION of EVENT))
        T]
        (\SHOWSEL SEL NIL T)))
      (T (TEDIT.PROMPTPRINT TEXTOBJ "Nothing to UNDO." T]))
)
```

(**TEDIT.UNDO.INSERTION**

[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 21-Apr-93 01:33 by jds

:: UNDO a prior Insert, Copy, or Include.

```
(PROG (OBJ DELETEDFN)
      (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL) ; Keep TEdit from reusing the current cache piece in the future --
      ; it is probably invalid
      (\DELETEDCH CH# (IPLUS CH# LEN))
)
```

```

      LEN TEXTOBJ)
(\FIXDLINES (fetch (TEXTOBJ LINES) of TEXTOBJ)
  (fetch (TEXTOBJ SEL) of TEXTOBJ)
  CH#
  (IPLUS CH# LEN)
  TEXTOBJ) ; Fix the line descriptors & selection
(TEDIT.UPDATE.SCREEN TEXTOBJ) ; Fix up the display for all this foofaraw
(replace (SELECTION POINT) of (fetch (TEXTOBJ SEL) of TEXTOBJ) with 'LEFT)
(\FIXSEL (fetch (TEXTOBJ SEL) of TEXTOBJ)
  TEXTOBJ) ; Really fix the selection
(replace THACTION of EVENT with 'Delete) ; Make the UNDO be UNDOable, by changing the event to a
; deletion.
)

```

(TEDIT.UNDO.DELETION

[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 21-Apr-93 12:01 by jds

;; UNDO a prior Deletion of text.

```

(PROG ((NPC (fetch (PIECE NEXTPIECE) of FIRSTPIECE))
  (PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
  (SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
  (OTEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))
  NEWPIECE INSPC OBJECT INSERTFN START-OF-PIECE)
  (SETQ INSPC (\CHTOPC CH# PCTB T))
  (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL) ; Keep future people from stepping on the current cache piece,
; which is probably no longer valid.

(COND
  ((fetch (TEXTOBJ TXTREADONLY) of TEXTOBJ) ; Don't change read-only documents.
  (RETURN)))
(COND
  ((IGREATERP CH# START-OF-PIECE)
  (SETQ INSPC (\SPLITPIECE INSPC (- CH# START-OF-PIECE)
    TEXTOBJ INSPC#)
  (SETQ NEWPIECE (create PIECE using FIRSTPIECE))
  (replace THFIRSTPIECE of EVENT with NEWPIECE)
  (bind (TL _ 0) while (ILESSP TL LEN) do (\INSERTPIECE NEWPIECE INSPC TEXTOBJ)
    ; Insert the piece back in
    (COND
      ([AND (SETQ OBJECT (fetch (PIECE POBJ) of NEWPIECE))
        (SETQ INSERTFN (IMAGEOBJPROP OBJECT 'WHENINSERTEDFN))
        ; If this is an imageobject, and it has an insertfn, call it.
        (APPLY* INSERTFN OBJECT (\TEDIT.PRIMARYW TEXTOBJ))
        NIL
        (TEXTSTREAM TEXTOBJ)
        (SETQ TL (IPLUS TL (fetch (PIECE PLEN) of FIRSTPIECE)))
        ; Keep track of how much we've re-inserted
        (SETQ FIRSTPIECE NPC)
        ; Move to the next piece to insert
        (AND NPC (SETQ NPC (fetch (PIECE NEXTPIECE) of NPC)))
        (SETQ NEWPIECE (create PIECE using FIRSTPIECE)))
        ; Done here because \INSERTPIECE creams the NEXTPIECE
        ; field.
        (replace (TEXTOBJ TEXTLEN) of TEXTOBJ with (IPLUS (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
          LEN)) ; Reset the text length and EOF ptr of the text stream.
        (\FIXILINES TEXTOBJ SEL CH# LEN OTEXTLEN) ; Fix the line descriptors & selection
        (TEDIT.UPDATE.SCREEN TEXTOBJ) ; Fix up the display for all this foofaraw
        (replace (SELECTION CH#) of SEL with CH#) ; Make the selection point at the re-inserted text
        (replace (SELECTION CHLIM) of SEL with (IPLUS CH# LEN))
        (replace (SELECTION DCH) of SEL with LEN)
        (replace (SELECTION POINT) of SEL with (fetch THPOINT of EVENT))
        (\TEDIT.SET.SEL.LOOKS SEL 'NORMAL)
        (\FIXSEL SEL TEXTOBJ) ; Really fix the selection
        (replace THACTION of EVENT with 'Insert) ; Make the UNDO be UNDOable, by changing the event to a
; insertion.
      )
    )
  )
)

```

)

(TEDIT.REDO

[LAMBDA (TEXTOBJ) ; Edited 30-May-91 21:27 by jds

;; REDO the last thing this guy did.

```

(PROG ((SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
  EVENT CH)
  (COND
    ((FETCH (TEXTOBJ TXTREADONLY) OF TEXTOBJ)
    ; The document is read-only; don't make any changes.
    NIL)
    ((SETQ EVENT (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ))
    ; There really is something to REDO Decide what, & do it.
    (RESETLST
      (RESETSAVE (CURSOR WAITINGCURSOR))
      (\SHOWSEL SEL NIL NIL)
      (SELECTQ (fetch THACTION of EVENT)

```

```

((Insert Copy Include) ; It was an insertion
  (TEDIT.REDO.INSERTION TEXTOBJ EVENT (IMAX 1 (SELECTQ (fetch (SELECTION POINT)
    of SEL)
    (LEFT (fetch (SELECTION CH#)
      of SEL))
    (RIGHT (fetch (SELECTION CHLIM)
      of SEL))
    NIL))))))
(Delete ; It was a deletion
  (\TEDIT.DELETE SEL TEXTOBJ))
(Replace ; It was a replacement (a del/insert combo)
  (TEDIT.REDO.REPLACE TEXTOBJ EVENT))
(LowerCase ; He lower-cased something
  (\TEDIT.LCASE.SEL TEXTOBJ TEXTOBJ SEL))
(UpperCase ; He upper-cased something
  (\TEDIT.LCASE.SEL TEXTOBJ TEXTOBJ SEL))
(Looks ; It was a looks change
  (TEDIT.REDO.LOOKS TEXTOBJ EVENT (IMAX 1 (SELECTQ (fetch (SELECTION POINT)
    of SEL)
    (LEFT (fetch (SELECTION CH#)
      of SEL))
    (RIGHT (fetch (SELECTION CHLIM)
      of SEL))
    NIL))))))
(ParaLooks ; It was a Paragraph looks change
  (TEDIT.REDO.PARALOOKS TEXTOBJ EVENT (IMAX 1 (SELECTQ (fetch (SELECTION POINT)
    of SEL)
    (LEFT (fetch (SELECTION CH#)
      of SEL))
    (RIGHT (fetch (SELECTION CHLIM)
      of SEL))
    NIL))))))
(Find ; EXACT-MATCH SEARCH COMMAND
  (RESETLST
    (RESETSAVE (CURSOR WAITINGCURSOR))
    (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)
    (SETQ SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
    (\SHOWSEL SEL NIL NIL)
    (SETQ CH (TEDIT.FIND TEXTOBJ (fetch THAUXINFO of EVENT)))
    (COND
      (CH (TEDIT.PROMPTPRINT TEXTOBJ "done.")
        (replace (SELECTION CH#) of SEL with CH)
        [replace (SELECTION CHLIM) of SEL
          with (IPLUS CH (NCHARS (fetch THAUXINFO of EVENT)
            (replace (SELECTION DCH) of SEL with (NCHARS (fetch THAUXINFO of EVENT)))
            (replace (SELECTION POINT) of SEL with 'RIGHT)
            (\FIXSEL SEL TEXTOBJ)
            (TEDIT.NORMALIZECARET TEXTOBJ)
            (\SHOWSEL SEL NIL T))
          (T (TEDIT.PROMPTPRINT TEXTOBJ "[Not found]"))))
        (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL)
        ; Drop the cached piece. WHY??
      )
    )
  (Move ReplaceMove) ; He moved some text
  (TEDIT.REDO.MOVE TEXTOBJ EVENT (fetch THLEN of EVENT)
    (IMAX 1 (SELECTQ (fetch (SELECTION POINT) of SEL)
      (LEFT (fetch (SELECTION CH#) of SEL))
      (RIGHT (fetch (SELECTION CHLIM) of SEL))
      NIL))
    (fetch THFIRSTPIECE of EVENT)))
(Get ; He did a GET -- not undoable.
  (TEDIT.PROMPTPRINT TEXTOBJ "You can't REDO a GET." T))
(Put ; He did a PUT -- not undoable.
  (TEDIT.PROMPTPRINT TEXTOBJ "You can't REDO a PUT." T))
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "REDO of the action " (fetch THACTION of EVENT)
  " isn't implemented."
  T))
(\SHOWSEL SEL NIL T))
(T (TEDIT.PROMPTPRINT TEXTOBJ "Nothing to REDO." T))

```

**(TEDIT.REDO.INSERTION**

```

[LAMBDA (TEXTOBJ EVENT CH#) ; Edited 3-Apr-95 15:55 by sybalsky:mv:envos
  ; REDO a prior Insert/Copy/Include of text.
  (PROG (INSPC INSPC# NPC (SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
    (PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
    (LEN (fetch THLEN of EVENT))
    (FIRSTPIECE (create PIECE using (CAR (fetch THFIRSTPIECE of EVENT))
      PNEW - T))
    (OTEXTLEN (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ))
    OBJ COPYFN ORIGFIRSTPC)
    (SETQ ORIGFIRSTPC FIRSTPIECE)
    (replace THFIRSTPIECE of EVENT with (LIST FIRSTPIECE)) ; So we can UNDO this, and remove the right set of pieces.
    (replace (TEXTOBJ \INSERTPCVALID) of TEXTOBJ with NIL) ; Force any further insertions to make new pieces.
    (SETQ NPC (fetch (PIECE NEXTPIECE) of FIRSTPIECE))
    (SETQ INSPC (\CHTOPC CH# PCTB T))
  )

```

```
[SETQ INSPC (COND
  ((IEQP CH# START-OF-PIECE) ; We're inserting just before an existing piece
   INSPC)
  (T (\SPLITPIECE INSPC (- CH# START-OF-PIECE)
    TEXTOBJ) ; We must split this piece, and insert before the second part.
  )
)
(bind (TL _ 0) while (ILESSP TL LEN) do ;; Loop thru the pieces of the prior insertion, inserting copies of enough of them to
;; cover the length of the insertion.
  [COND
    ((SETQ OBJ (fetch (PIECE POBJ) of FIRSTPIECE))
     ; This piece describes an object
    )
    [COND
      [(SETQ COPYFN (IMAGEOBJPROP OBJ 'COPYFN))
       (SETQ OBJ (APPLY* COPYFN OBJ (fetch (TEXTOBJ STREAMHINT)
         of TEXTOBJ)
         (fetch (TEXTOBJ STREAMHINT)
         of TEXTOBJ)))
      ]
      (COND
        ((EQ OBJ 'DON'T)
         (TEDIT.PROMPTPRINT TEXTOBJ "COPY of this object not
           allowed." T)
         (RETFROM 'TEDIT.COPY))
        (T (replace (PIECE POBJ) of FIRSTPIECE with OBJ)
         (OBJ (replace (PIECE POBJ) of FIRSTPIECE
           with (COPY OBJ)
         )
        )
      )
      (COND
        ((SETQ COPYFN (IMAGEOBJPROP OBJ 'WHENCOPIEDFN))
         ; If there's an eventfn for copying, use it.
         (APPLY* COPYFN OBJ (WINDOWPROP
           (CAR (fetch (TEXTOBJ \WINDOW)
             of TEXTOBJ))
           'DSP)
         (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
         (fetch (TEXTOBJ STREAMHINT) of TEXTOBJ)
         (\INSERTPIECE FIRSTPIECE INSPC TEXTOBJ)
         ; Insert the piece back in
         (SETQ TL (IPLUS TL (fetch (PIECE PLEN) of FIRSTPIECE)))
         ; Keep track of how much we've re-inserted
         (SETQ FIRSTPIECE (create PIECE using NPC PNEW _ T))
         ; Move to the next piece to insert
         (AND NPC (SETQ NPC (fetch (PIECE NEXTPIECE) of NPC)))
         ; Done here because \INSERTPIECE creams the NEXTPIECE
         ; field.
        )
      )
    )
  (\TEDIT.DIFFUSE.PARALOOKS (fetch (PIECE PREVPIECE) of ORIGFIRSTPC)
   INSPC) ; propagate paragraph formatting into the new insertion
  (replace (TEXTOBJ TEXTLEN) of TEXTOBJ with (IPLUS (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)
   LEN)) ; Reset the text length and EOF ptr of the text stream.
  (\FIXILINES TEXTOBJ SEL CH# LEN OTEXTLEN) ; Fix the line descriptors & selection
  (TEDIT.UPDATE.SCREEN TEXTOBJ) ; Fix up the display for all this foofaraw
  (replace (SELECTION CH#) of SEL with CH#) ; Make the selection point at the re-inserted text
  (replace (SELECTION CHLIM) of SEL with (IPLUS CH# LEN))
  (replace (SELECTION DCH) of SEL with LEN)
  (\TEDIT.SET.SEL.LOOKS SEL 'NORMAL)
  (\FIXSEL SEL TEXTOBJ) ; Really fix the selection
  (replace THACTION of EVENT with 'Insert) ; Make the UNDO be UNDOable, by changing the event to a
; insertion.
)
]
```

**(TEDIT.UNDO.MOVE**

```
[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 30-May-91 21:27 by jds
; UNDO a MOVE command
  (PROG ((TOOBJ (fetch THAUXINFO of EVENT))
    (FROMOBJ (fetch THTEXTOBJ of EVENT))
    (SOURCECH# (fetch THOLDINFO of EVENT))
    (CH# (fetch THCH# of EVENT))
    TOSEL TOTEXTLEN)
    (\SHOWSEL (fetch (TEXTOBJ SEL) of TOOBJ)
     NIL NIL) ; Turn off the selections in the old source and target documents
    (\SHOWSEL (fetch (TEXTOBJ SEL) of FROMOBJ)
     NIL NIL)
    (\DELETECH CH# (IPLUS CH# LEN)
     LEN FROMOBJ) ; Delete the characters we moved, from the place we moved
; them to
    (\FIXDLINES (fetch (TEXTOBJ LINES) of FROMOBJ)
     (fetch (TEXTOBJ SEL) of FROMOBJ)
     CH#
     (IPLUS CH# LEN)
     FROMOBJ)
    (replace (SELECTION CH#) of (fetch (TEXTOBJ SEL) of FROMOBJ) with (replace (SELECTION CHLIM)
     of (fetch (TEXTOBJ SEL) of FROMOBJ)
     with CH#))
    ; Make this document's selection be a point sel at the place the
; text used to be.
  )
]
```

```
(replace (SELECTION DCH) of (fetch (TEXTOBJ SEL) of FROMOBJ) with 0)
(replace (SELECTION POINT) of (fetch (TEXTOBJ SEL) of FROMOBJ) with 'LEFT)
(SETQ TOTEXTLEN (fetch (TEXTOBJ TEXTLEN) of TOOBJ)) ; Mark lines for update, and fix the selection
; The pre-insertion len of the place the text is returning to, for the
; line updater below
(\TEDIT.INSERT.PIECES TOOBJ SOURCECH# (fetch THFIRSTPIECE of EVENT)
LEN)
```

:: Put the pieces we moved back where they came from (no need to copy them, since we did that on the original move.)

```
(\FIXILINES TOOBJ (fetch (TEXTOBJ SEL) of TOOBJ)
SOURCECH# LEN TOTEXTLEN) ; Mark lines that need updating, and fix up the selection
(add (fetch (TEXTOBJ TEXTLEN) of TOOBJ)
LEN) ; Update the text length of the erstwhile move source
(TEDIT.UPDATE.SCREEN FROMOBJ) ; Update the erstwhile text location's image.
(COND
((NEQ FROMOBJ TOOBJ) ; If they aren't the same document, we need to update the other
; document image as well.
(TEDIT.UPDATE.SCREEN TOOBJ)))
(\FIXSEL (fetch (TEXTOBJ SEL) of TOOBJ)
TOOBJ) ; Fix up the selections so their images will be OK
(\FIXSEL (fetch (TEXTOBJ SEL) of FROMOBJ)
FROMOBJ)
(\COPYSEL (fetch (TEXTOBJ SEL) of FROMOBJ)
TEDIT.SELECTION) ; It's handy to think of this as the last selection made, also.
(replace THACTION of EVENT with 'Move)
(replace THTEXTOBJ of EVENT with TOOBJ)
(replace THAUXINFO of EVENT with FROMOBJ)
(replace THOLDINFO of EVENT with CH#)
(replace THCH# of EVENT with SOURCECH#)
(\SHOWSEL (fetch (TEXTOBJ SEL) of TOOBJ)
NIL T)
(\SHOWSEL (fetch (TEXTOBJ SEL) of FROMOBJ)
NIL T])
```

**(TEDIT.UNDO.REPLACE**

```
[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 22-Mar-95 16:47 by sybalsky:mv:envos
(PROG ((OLDEVENT (fetch THOLDINFO of EVENT))
(CH# (fetch THCH# of EVENT))
(SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)))
(\SHOWSEL SEL NIL NIL)
(TEDIT.UNDO.INSERTION TEXTOBJ EVENT LEN CH# FIRSTPIECE)
(\SHOWSEL SEL NIL NIL)
(TEDIT.UNDO.DELETION TEXTOBJ OLDEVENT (fetch THLEN of OLDEVENT)
CH#
(CAR (fetch THFIRSTPIECE of OLDEVENT)))
(replace THOLDINFO of OLDEVENT with EVENT)
(replace THACTION of OLDEVENT with 'Replace)
(replace THOLDINFO of EVENT with NIL)
(\TEDIT.HISTORYADD TEXTOBJ OLDEVENT)
(replace (SELECTION CH#) of SEL with CH#)
(replace (SELECTION CHLIM) of SEL with (IPLUS CH# (fetch THLEN of OLDEVENT)))
(replace (SELECTION DCH) of SEL with (fetch THLEN of OLDEVENT))
(replace (SELECTION POINT) of SEL with (fetch THPOINT of EVENT))
(replace THPOINT of OLDEVENT with (fetch THPOINT of EVENT))
(\FIXSEL SEL TEXTOBJ)
(\SHOWSEL SEL NIL T])
```

**(TEDIT.REDO.REPLACE**

```
[LAMBDA (TEXTOBJ EVENT) ; Edited 30-May-91 21:28 by jds
(PROG ((OLDEVENT (fetch THOLDINFO of EVENT))
(CH# (fetch (SELECTION CH#) of (fetch (TEXTOBJ SEL) of TEXTOBJ)))
(SEL (fetch (TEXTOBJ SEL) of TEXTOBJ)))
(\SHOWSEL SEL NIL NIL)
(\DELETECH (fetch (SELECTION CH#) of SEL)
(fetch (SELECTION CHLIM) of SEL)
(IDIFERENCE (fetch (SELECTION CHLIM) of SEL)
(fetch (SELECTION CH#) of SEL))
TEXTOBJ)
(\FIXDLINES (fetch (TEXTOBJ LINES) of TEXTOBJ)
SEL
(fetch (SELECTION CH#) of SEL)
(fetch (SELECTION CHLIM) of SEL)
TEXTOBJ)
(replace (SELECTION POINT) of SEL with 'LEFT)
(TEDIT.REDO.INSERTION TEXTOBJ EVENT CH#)
(replace THOLDINFO of EVENT with (SETQ OLDEVENT (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ)))
(replace THACTION of OLDEVENT with 'Replace)
(replace THACTION of EVENT with 'Replace)
(replace THCH# of EVENT with CH#)
(\TEDIT.HISTORYADD TEXTOBJ EVENT])
```

**(TEDIT.REDO.MOVE**

```
[LAMBDA (TEXTOBJ EVENT LEN CH# FIRSTPIECE) ; Edited 30-May-91 21:28 by jds
(PROG ((FROMOBJ TEXTOBJ)
```

```
(SOURCECH# (fetch THOLDINFO of EVENT))
(OLDCH# (fetch THCH# of EVENT))
(SEL (fetch (TEXTOBJ SEL) of TEXTOBJ))
(MOVESEL (fetch (TEXTOBJ MOVESEL) of TEXTOBJ))
OLDCHLIM)
(replace (SELECTION CH#) of MOVESEL with OLDCH#)
(replace (SELECTION CHLIM) of MOVESEL with (IPLUS OLDCH# LEN))
(replace (SELECTION DCH) of MOVESEL with LEN)
(replace (SELECTION SET) of MOVESEL with T)
(\FIXSEL MOVESEL TEXTOBJ)
(\TEDIT.SET.SEL.LOOKS MOVESEL 'MOVE)
(TEDIT.MOVE MOVESEL SEL])
```

)

(PUTPROPS TEDITHISTORY COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1990 1991 1993 1995))

**FUNCTION INDEX**

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

---

**VARIABLE INDEX**

TEDIT.HISTORY.TYPELST ...	1	TEDIT.HISTORYLST .....	1
---------------------------	---	------------------------	---

---

**CONSTANT INDEX**

\SCRATCHLEN .....

---