```
(RPAQQ LAFITE-TEDITCOMS (;; Lafite's more explicit dependencies on "internals" of TEDIT
                         (FNS LA.ADJUST.FORMATTING LA.DETACH.TEDIT TEDIT.ASSURE.NO.BACKING.FILE
                              LA.WINDOW.FROM.TEXTSTREAM)
                         (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (SOURCE)
                                                                 LAFITE-DECLS)
                               (GLOBALVARS *TEDIT-FILE-READTABLE*)
                               (LOCALVARS . T))))
```

;; Lafite's more explicit dependencies on "internals" of TEDIT

```
(DEFINEQ
```

(LA.ADJUST.FORMATTING

```
[LAMBDA (FORMATSTREAM OUTSTREAM BYTE-LENGTHS)           ; Edited 18-Jan-2024 10:33 by rmk
                                                       ; Edited 13-Jan-2024 21:53 by rmk
                                                       ; Edited  3-Jun-88 18:24 by bvm
```

;; Adjusts the formatting info FORMATSTREAM to account for the prepending of one or more "paragraphs" of default looking text, whose lengths
;; are given by BYTE-LENGTHS (or a single number if just one piece). It then writes the resulting formatting to OUTSTREAM.

;; The original code assumed that there were pieces at the beginning of the file that represented the pageframes and charlooks and paralooks
;; vectors, and that that the information for new pieces had to come just after the last of those. That required parsing the initial pieces, with too
;; much knowledge of Tedit internals. The reading function\TEDIT.GET.PIECES3 has been revised to relax that ordering constraint, so it is now
;; possible just to slap the new information on the front and place an updated trailer at the end.

```
(PROG ((TRAILER (\TEDIT.GET.TRAILER FORMATSTREAM))
        PIECESTART PCCOUNT TRAILERSIZE)
      (CL:UNLESS (EQ 3 (pop TRAILER))
             (RETURN NIL))
      (SETQ PIECESTART (pop TRAILER))
      (SETQ TRAILERSIZE (pop TRAILER))
      (SETQ PCCOUNT (CADR TRAILER))                    ; Skip version
```

;; Write the new-piece information at the beginning of OUTSTREAM.

;; This code allows for multiple inserted pieces, but unfortunately if the textstream already has any paragraph formatting, we can't make the pieces
;; be different paragraphs without making them also have non-default paralooks.

;; The original code used the looks index of the first real piece, 1 if none was encountered. That was arbitrary, here we arbitrarily assign whatever
;; charlooks got the first index. (Tedit could arrange for that also to be the index of the first piece, if it mattered).

```
      (add PIECESTART (for PIECELEN inside BYTE-LENGTHS sum (\TEDIT.PUT.CHARLOOKS1 OUTSTREAM PIECELEN 1)
                                                            (add PCCOUNT 1)
                                                            PIECELEN))
```

;; We are looking just at the format part so presume that its piece descriptions start at its 0. Copy rest of piece info, not including the old trailer.

```
      (COPYBYTES FORMATSTREAM OUTSTREAM 0 (IDIFFERENCE (GETEOFPTR FORMATSTREAM)
                                                       TRAILERSIZE))
```

;; The piece-pointer for the new trailer is adjusted above to account for the new pieces in the corresponding CHARSTREAM.

```
      (\TEDIT.PUT.TRAILER OUTSTREAM PIECESTART PCCOUNT 3)
      (RETURN OUTSTREAM])
```

(LA.DETACH.TEDIT

```
[LAMBDA (TEXTSTREAM)                                    ; Edited 14-Jan-2024 12:56 by rmk
                                                       ; Edited  3-Jun-88 17:27 by bvm
```

;; Removes the TEXTSTREAM from the window, if any, it is being edited in.

```
(TEXTPROP TEXTSTREAM '\WINDOW NIL])
```

(TEDIT.ASSURE.NO.BACKING.FILE

```
[LAMBDA (TEXTSTREAM)                                    ; Edited 13-Jan-2024 18:08 by rmk
                                                       ; Edited 18-Jun-2023 09:31 by rmk
                                                       ; Edited 29-Oct-2022 22:34 by rmk
                                                       ; Edited 20-May-92 11:25 by rmk:
```

;; This puts the contents of TEXTSTREAM to a nodircore file (if it isn't already on nodircore), and then sets it up for continuing in the current editing
;; session. Essentially eliminates the file-system backing store.

```
(LET* ((TEXTOBJ (TEXTOBJ TEXTSTREAM))
       (OFILE (GETTOBJ TEXTOBJ TXTFILE))
        NEWFILE)
      (CL:WHEN [AND (TYPE? STREAM OFILE)
                    (NEQ 'NODIRCORE (FETCH (FDEV DEVICENAME) OF (FETCH (STREAM DEVICE) OF (TRUEFILENAME
                                                                                                 OFILE]
```

```
                    (SETQ NEWFILE (OPENSTREAM '{NODIRCORE} 'BOTH))
```
        ;; \TEDIT.PUT.PCTB will save the current text and looks in NEWFILE, leaving it open. It returns the sequence of new looks for
        ;; continued editing, where all the file pieces point to their position in NEWFILE. But the file PCONTENTS do not yet point to the new
        ;; stream.
```
                    (CLOSEF? OFILE)
                    (\TEDIT.INSERT.NEWPIECES NEWFILE TEXTOBJ (\TEDIT.PUT.PCTB TEXTOBJ NEWFILE NIL T))
                    (FSETTOBJ TEXTOBJ TXTFILE NIL)
                    (PUTTEXTPROP TEXTOBJ 'CACHE NEWFILE)
                    TEXTSTREAM)])
```

(**LA.WINDOW.FROM.TEXTSTREAM**

```
  [LAMBDA (TEXTSTREAM)                                  ; Edited 18-Jun-2023 09:09 by rmk
                                                        ; Edited 23-Sep-87 15:36 by bvm:

    (\TEDIT.MAINW TEXTSTREAM])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (SOURCE)
        LAFITE-DECLS)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *TEDIT-FILE-READTABLE*)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)
```

FUNCTION INDEX