

File created: 20-Oct-2023 00:11:10 {WMEDLEY}<lispusers>tedit-process-killer.;2

edit by: rmk

changes to: (FNS TEDIT-PROCESS-P)

previous date: 9-Mar-89 15:01:15 {WMEDLEY}<lispusers>tedit-process-killer.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;  
;; Copyright (c) 1987-1989 by Xerox Corporation.
```

(RPAQQ **TEDIT-PROCESS-KILLERCOMS**

```
[;; This package provides various ways to kill tedit processes. Using START-TEDIT-KILLER, one can keep the total number of tedit processes  
;; under the threshold TEDIT-PROCESS-LIMIT. One can also call KILL-PROCESS-OF-TEDIT-WINDOW to kill the Tedit processes for a  
;; given window and its attached windows.
```

```
(GLOBALVARS TEDIT-PROCESS-LIMIT TEDIT-KILLER-WAIT-TIME TEDIT-PROCESSES TEDIT-CREATION-TIME)
```

;;; These two vars are advertised.

```
(INITVARS (TEDIT-PROCESS-LIMIT 5)  
          (TEDIT-KILLER-WAIT-TIME 10000))  
(VARS (TEDIT-PROCESSES NIL)  
      (TEDIT-CREATION-TIME NIL))
```

;;; Here are the advertised functions.

```
(FNS START-TEDIT-KILLER STOP-TEDIT-KILLER KILL-PROCESS-OF-TEDIT-WINDOW RESTART-PROCESS-OF-TEDIT-WINDOW  
    WITHOUT-TEDIT-PROCESS)
```

;;; The rest of these are internal.

```
(FNS TEDIT-KILLER \TEDIT.BUTTONEVENTFN-BEFORE-ADVICE)  
(FNS MARK-AS-WITHOUT-PROCESS UNMARK-AS-WITHOUT-PROCESS WITHOUT-PROCESS)  
(FNS ALL-TEDIT-PROCESSES TEDIT-PROCESS-P KILL-PROCESS-OF-TEDIT-WINDOW1 KILL-TEDIT-PROCESS  
    MAKE-NEW-TEDIT-PROCESS RESTART-PROCESS-OF-TEDIT-WINDOW1 TEDIT-KILLER-CLEANUP)
```

;;; NOTE: this advising smashes whatever advice was previously on these functions!

```
(DECLARE%: DONTEVAL@LOAD DOCOPY (ADVISE \TEDIT.QUIT TEDIT \TEDIT.BUTTONEVENTFN  
                                ;; PROCESS.APPLY advice is mainly for NoteCards - fixes a misuse of this function that makes it  
                                ;; impossible to use monitors inside a TEdit menu fn.  
                                (PROCESS.APPLY :IN \TEDIT.BUTTONEVENTFN)  
                                (PROCESSP :IN TEDIT.DEACTIVATE.WINDOW)  
                                (PROCESSP :IN \TEDIT.ACTIVE.WINDOWP))  
    (P (START-TEDIT-KILLER)))
```

```
;; This package provides various ways to kill tedit processes. Using START-TEDIT-KILLER, one can keep the total number of tedit processes under the  
;; threshold TEDIT-PROCESS-LIMIT. One can also call KILL-PROCESS-OF-TEDIT-WINDOW to kill the Tedit processes for a given window and its  
;; attached windows.
```

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

```
(GLOBALVARS TEDIT-PROCESS-LIMIT TEDIT-KILLER-WAIT-TIME TEDIT-PROCESSES TEDIT-CREATION-TIME)  
)
```

;;; These two vars are advertised.

(RPAQ? **TEDIT-PROCESS-LIMIT** 5)

(RPAQ? **TEDIT-KILLER-WAIT-TIME** 10000)

(RPAQQ **TEDIT-PROCESSES** NIL)

(RPAQQ **TEDIT-CREATION-TIME** NIL)

;;; Here are the advertised functions.

```
(DEFINEQ
```

(**START-TEDIT-KILLER**

```
[LAMBDA NIL  
  (DECLARE (GLOBALVARS TEDIT-CREATION-TIME TEDIT-PROCESSES)) ; Edited 11-Dec-87 19:43 by Randy.Gobbel
```

```
;; Kill off old killers.
(STOP-TEDIT-KILLER)
;; Reset stuff and start er up.
(SETQ TEDIT-CREATION-TIME (CLOCK 0))
(SETQ TEDIT-PROCESSES (ALL-TEDIT-PROCESSES))
(ADD.PROCESS ' (TEDIT-KILLER)
  'RESTARTABLE
  'HARDRESET))
```

(STOP-TEDIT-KILLER

[LAMBDA NIL

; Edited 2-Feb-88 14:08 by Randy.Gobbel

```
;; Kill off old killers.
(DECLARE (GLOBALVARS \PROCESSES))
(for P in \PROCESSES when [EQ 'TEDIT-KILLER (CAR (PROCESSPROP P 'FORM)]
  do (DEL.PROCESS P)
  (until (NOT (PROCESSP P)) do (BLOCK]))
```

(KILL-PROCESS-OF-TEDIT-WINDOW

[LAMBDA (WINDOW)

; Edited 11-Dec-87 19:17 by Randy.Gobbel

```
;; Kill the process of this window, and anybody who is attached to me (recursively).
(KILL-PROCESS-OF-TEDIT-WINDOW1 (MAINWINDOW WINDOW))
```

(RESTART-PROCESS-OF-TEDIT-WINDOW

[LAMBDA (WINDOW)

(* SCB%: " 2-May-86 10:41")

(* Move down the attached windows tree from the mainwindow, firing up a new process for each TEdit.)

```
(RESTART-PROCESS-OF-TEDIT-WINDOW1 (MAINWINDOW WINDOW))
(TTY.PROCESS (WINDOWPROP (MAINWINDOW WINDOW)
  'PROCESS))
```

(WITHOUT-TEDIT-PROCESS

[LAMBDA (WINDOW)

(* SCB%: " 2-May-86 16:08")

(EQ 'TEDIT (**WITHOUT-PROCESS** WINDOW))

)

;;; The rest of these are internal.

(DEFINEQ

(TEDIT-KILLER

[LAMBDA NIL

; Edited 11-Dec-87 20:53 by Randy.Gobbel

```
(DECLARE (GLOBALVARS TEDIT-PROCESSES TEDIT-KILLER-WAIT-TIME TEDIT-CREATION-TIME TEDIT-PROCESS-LIMIT))
(while T bind (TO-KILL _ 0) do (DISMISS TEDIT-KILLER-WAIT-TIME)
  (if (AND TEDIT-PROCESSES (ILESSP TEDIT-KILLER-WAIT-TIME (IDIFFERENCE
    (CLOCK 0)
    TEDIT-CREATION-TIME)))
    then (SETQ TEDIT-PROCESSES (for P in TEDIT-PROCESSES
      when (TEDIT-PROCESS-P P) collect P))
      (SETQ TO-KILL (IDIFFERENCE (LENGTH TEDIT-PROCESSES)
        TEDIT-PROCESS-LIMIT)))
    ;; Kill processes, starting from the least recently used.
    (until (ILEQ TO-KILL 0) for P in (REVERSE TEDIT-PROCESSES)
      do (COND
        ((AND (NEQ (TTY.PROCESS)
          P)
          (PROCESSP P))
          (KILL-TEDIT-PROCESS P)
          (SETQ TO-KILL (SUB1 TO-KILL]))
```

(\TEDIT.BUTTONEVENTFN-BEFORE-ADVICE

[LAMBDA (W)

; Edited 11-Dec-87 19:45 by Randy.Gobbel

```
(DECLARE (GLOBALVARS TEDIT-PROCESSES))
(LET [(TEXTOBJ (TEXTOBJ W))
  (PROCESS (WINDOWPROP W 'PROCESS)]
  (COND
    ([AND TEXTOBJ (NOT (PROCESSP PROCESS))
      (MOUSESTATE (OR LEFT MIDDLE))
      (NOT (TEXTPROP TEXTOBJ 'READONLY))
      (NOT (SHIFTDOWNP 'SHIFT))
      (NOT (SHIFTDOWNP 'CTRL))
      (NOT (SHIFTDOWNP 'META))
      (NOT (KEYDOWNP 'MOVE))
      (NOT (KEYDOWNP 'COPY))
      (SETQ PROCESS (MAKE-NEW-TEDIT-PROCESS W))
```

```

      (SETQ TEDIT-PROCESSES (CONS PROCESS TEDIT-PROCESSES))
      (TTY.PROCESS PROCESS))
    ([AND (PROCESSP PROCESS)
      (NOT (EQ PROCESS (CAR TEDIT-PROCESSES)
                    ; We're using the process, so move it to the front of the list.
      (SETQ TEDIT-PROCESSES (CONS PROCESS (DREMOVE PROCESS TEDIT-PROCESSES))
    )

```

```
(DEFINEQ
```

(MARK-AS-WITHOUT-PROCESS

```

  [LAMBDA (WINDOW PROCESS-TYPE)
    (WINDOWPROP WINDOW 'WITHOUT-PROCESS PROCESS-TYPE)]
  (* SCB%: " 2-May-86 15:44")

```

(UNMARK-AS-WITHOUT-PROCESS

```

  [LAMBDA (WINDOW)
    (WINDOWPROP WINDOW 'WITHOUT-PROCESS NIL)]
  (* SCB%: " 2-May-86 15:44")

```

(WITHOUT-PROCESS

```

  [LAMBDA (WINDOW)
    (WINDOWPROP WINDOW 'WITHOUT-PROCESS)]
  (* SCB%: " 2-May-86 15:43")

```

```
)
```

```
(DEFINEQ
```

(ALL-TEDIT-PROCESSES

```

  [LAMBDA NIL
    (* * Gather all the extant tedit processes.)

```

```

  (DECLARE (GLOBALVARS \PROCESSES))
  (for P in \PROCESSES when (TEDIT-PROCESS-P P) collect P])

```

(TEDIT-PROCESS-P

```

  [LAMBDA (PROCESS)
    ; Edited 20-Oct-2023 00:11 by rmk
    ; Edited 2-Feb-88 14:15 by Randy.Gobbel

    ;; rg 2/2/88: Now looks at process name instead of checking TTYENTRYFN = \TEDIT.PROCENTRYFN, which failed for TEdits that had never been
    ;; given the TTY (look at \TEDIT.COMMAND.LOOP code). This will miss processes that have been given another name, but works in practice for
    ;; all cases that I know of.

    (AND (PROCESSP PROCESS)
      (EQ (STRPOS "Tedit" (PROCESSPROP PROCESS 'NAME))
        1)
      (EQ (CAR (PROCESSPROP PROCESS 'FORM))
        '\TEDIT1])

```

(KILL-PROCESS-OF-TEDIT-WINDOW1

```

  [LAMBDA (WINDOW)
    (LET [(PROCESS (WINDOWPROP WINDOW 'PROCESS)
      (AND (TEDIT-PROCESS-P PROCESS)
        (KILL-TEDIT-PROCESS PROCESS))
      (for W in (ATTACHEDWINDOWS WINDOW) do (KILL-PROCESS-OF-TEDIT-WINDOW1 W))
    (* SCB%: " 1-May-86 17:37")

```

(KILL-TEDIT-PROCESS

```

  [LAMBDA (PROCESS)
    ; Edited 11-Dec-87 20:06 by Randy.Gobbel

    ;; Save the state that Tedit bashes, and then kill the process. Only TEdits have TEXTOBJS, so this won't go killing other kinds of processes. Won't
    ;; kill if the TEdit is in the middle of an operation.

    ;; rrp 10/19/87: Now also saves TXTFILE property.

    (LET* [(WINDOW (PROCESSPROP PROCESS 'WINDOW))
      (TEXTOBJ (WINDOWPROP WINDOW 'TEXTOBJ))
      (COND
        ((AND (WINDOWP WINDOW)
          TEXTOBJ
          (NOT (fetch (TEXTOBJ EDITOPACTIVE) of TEXTOBJ)))
          (WINDOWPROP WINDOW 'TXTHISTORY (fetch (TEXTOBJ TXTHISTORY) of TEXTOBJ))
          (WINDOWPROP WINDOW 'TXTFILE (fetch (TEXTOBJ TXTFILE) of TEXTOBJ))
          (WINDOWPROP WINDOW 'SELWINDOW (fetch (TEXTOBJ SELWINDOW) of TEXTOBJ))
          (WINDOWPROP WINDOW 'SAVEDPROCFORM (PROCESSPROP PROCESS 'FORM))
          (WINDOWPROP WINDOW 'SAVEDRESTARTFORM (PROCESSPROP PROCESS 'RESTARTFORM))
          (WINDOWPROP WINDOW 'SAVEDRESTARTABLE (PROCESSPROP PROCESS 'RESTARTABLE))
          (WINDOWPROP WINDOW 'SAVEDPROCNAME (PROCESSPROP PROCESS 'NAME))

          ;; Mark the window so we know we can restart the process. Atomic action to turn off the process.

          (UNINTERRUPTABLY
            (MARK-AS-WITHOUT-PROCESS WINDOW 'TEDIT)
            (DEL.PROCESS PROCESS)))]))

```

(MAKE-NEW-TEDIT-PROCESS

```

  [LAMBDA (WINDOW)
    ; Edited 9-Mar-89 14:58 by Randy.Gobbel

```

```

;; This assumes that WINDOW really is the window of a restartable TEdit.
;; Build a new TEdit process recovering saved PROCESSPROPS from the window.
;; rht 2/9/87: Added a check that SAVEDPROCFORM of WINDOW is non-nil in case WINDOW just got smashed.
;; rht&sb 4/24/87: Now smashes windowprops after reading them by calling TEDIT-KILLER-CLEANUP.
;; rrp 10/19/87: Now restores TXTFILE property as well.
(LET ((TEXTOBJ (TEXTOBJ WINDOW))
      (TXTFILE (WINDOWPROP WINDOW 'TXTFILE))
      (PROCESS SAVEDPROCFORM))
  (replace (TEXTOBJ TXTHISTORY) of TEXTOBJ with (WINDOWPROP WINDOW 'TXTHISTORY))
  (replace (TEXTOBJ SELWINDOW) of TEXTOBJ with (WINDOWPROP WINDOW 'SELWINDOW))
  [if (AND TXTFILE (NOT (STREQUAL TXTFILE "")))
      then (replace (TEXTOBJ TXTFILE) of TEXTOBJ with (OPENSTREAM (FULLNAME TXTFILE)
                                                                    'INPUT
                                                                    'OLD])

;; Atomic action to restore the process.
(if (SETQ SAVEDPROCFORM (WINDOWPROP WINDOW 'SAVEDPROCFORM))
    then (UNINTERRUPTABLY
          [SETQ PROCESS (ADD.PROCESS SAVEDPROCFORM 'NAME (LET* ((PROCNAME (WINDOWPROP WINDOW
                                                                    'SAVEDPROCNAME))
                                                                    (POS (STRPOS "#" PROCNAME)))
                                                                    (OR (SUBSTRING PROCNAME 1
                                                                    (AND POS (SUB1 POS)))
                                                                    PROCNAME))
                                                                    'RESTARTABLE
                                                                    (WINDOWPROP WINDOW 'SAVEDRESTARTABLE)
                                                                    'RESTARTFORM
                                                                    (WINDOWPROP WINDOW 'SAVEDRESTARTFORM)
                                                                    (TEDIT-KILLER-CLEANUP WINDOW)
                                                                    (PROCESSPROP PROCESS 'WINDOW WINDOW)
                                                                    (WINDOWPROP WINDOW 'PROCESS PROCESS)))
          PROCESS])

PROCESS])

```

(RESTART-PROCESS-OF-TEDIT-WINDOW1

```

[LAMBDA (WINDOW)
  (* SCB%: " 2-May-86 16:09")
  (* Only restart this guy if he used to have a TEdit process.)
  (AND (WITHOUT-TEDIT-PROCESS WINDOW)
        (MAKE-NEW-TEDIT-PROCESS WINDOW))
  (for W in (ATTACHEDWINDOWS WINDOW) do (RESTART-PROCESS-OF-TEDIT-WINDOW1 W))

```

(TEDIT-KILLER-CLEANUP

```

[LAMBDA (WINDOW) ; Edited 11-Dec-87 20:13 by Randy.Gobbel

```

```

;; This unmarks the window and also throws away any cruft we left on windowprops.

```

```

;; rrp 10/19/87: Now trashes TXTFILE property as well.

```

```

(WINDOWPROP WINDOW 'TXTHISTORY NIL)
(WINDOWPROP WINDOW 'TXTFILE NIL)
(WINDOWPROP WINDOW 'SELWINDOW NIL)
(WINDOWPROP WINDOW 'SAVEDPROCFORM NIL)
(WINDOWPROP WINDOW 'SAVEDPROCNAME NIL)
(WINDOWPROP WINDOW 'SAVEDRESTARTABLE NIL)
(WINDOWPROP WINDOW 'SAVEDRESTARTFORM NIL)
(UNMARK-AS-WITHOUT-PROCESS WINDOW])

```

```

)

```

```

;;; NOTE: this advising smashes whatever advice was previously on these functions!

```

```

(DECLARE%: DONTVAL@LOAD DOCOPY

```

```

[XCL:REINSTALL-ADVICE '\TEDIT.QUIT :AFTER '(:LAST (UNMARK-AS-WITHOUT-PROCESS W]

```

```

[XCL:REINSTALL-ADVICE 'TEDIT :BEFORE '(:LAST (SETQ TEDIT-CREATION-TIME (CLOCK 0)
: AFTER
' (:LAST (SETQ TEDIT-PROCESSES (CONS !VALUE TEDIT-PROCESSES]

```

```

[XCL:REINSTALL-ADVICE '\TEDIT.BUTTONEVENTFN :BEFORE '(:LAST (\TEDIT.BUTTONEVENTFN-BEFORE-ADVICE W]

```

```

[XCL:REINSTALL-ADVICE ' (PROCESS.APPLY :IN \TEDIT.BUTTONEVENTFN)
: AROUND
' (:LAST (ADD.PROCESS (LIST USERFN (KWOTE W]

```

```

[XCL:REINSTALL-ADVICE ' (PROCESSP :IN TEDIT.DEACTIVATE.WINDOW)
: AFTER
' (:LAST (RETURN (OR !VALUE (WITHOUT-TEDIT-PROCESS (STKARG 'W 'TEDIT.DEACTIVATE.WINDOW]

```

```

[XCL:REINSTALL-ADVICE ' (PROCESSP :IN \TEDIT.ACTIVE.WINDOWP)
: AFTER
' (:LAST (RETURN (OR !VALUE (WITHOUT-TEDIT-PROCESS (STKARG 'W '\TEDIT.ACTIVE.WINDOWP]

```

```

(READWISE \TEDIT.QUIT TEDIT \TEDIT.BUTTONEVENTFN (PROCESS.APPLY :IN \TEDIT.BUTTONEVENTFN)

```

```
{MEDLEY}<notecards>lispusers>TEDIT-PROCESS-KILLER.;1
```

Page 5

```
(PROCESSP :IN TEDIT.DEACTIVATE.WINDOW)  
(PROCESSP :IN \TEDIT.ACTIVE.WINDOWP))
```

```
(START-TEDIT-KILLER)  
)
```

```
(PUTPROPS TEDIT-PROCESS-KILLER COPYRIGHT ("Xerox Corporation" 1987 1988 1989))
```

FUNCTION INDEX

ALL-TEDIT-PROCESSES	3	STOP-TEDIT-KILLER	2
KILL-PROCESS-OF-TEDIT-WINDOW	2	TEDIT-KILLER	2
KILL-PROCESS-OF-TEDIT-WINDOW1	3	TEDIT-KILLER-CLEANUP	4
KILL-TEDIT-PROCESS	3	TEDIT-PROCESS-P	3
MAKE-NEW-TEDIT-PROCESS	3	UNMARK-AS-WITHOUT-PROCESS	3
MARK-AS-WITHOUT-PROCESS	3	WITHOUT-PROCESS	3
RESTART-PROCESS-OF-TEDIT-WINDOW	2	WITHOUT-TEDIT-PROCESS	2
RESTART-PROCESS-OF-TEDIT-WINDOW1	4	\TEDIT.BUTTONEVENTFN-BEFORE-ADVICE	2
START-TEDIT-KILLER	1		

ADVICE INDEX

TEDIT	4	(PROCESS.APPLY :IN \TEDIT.BUTTONEVENTFN)	4
\TEDIT.BUTTONEVENTFN	4	(PROCESSP :IN TEDIT.DEACTIVATE.WINDOW)	4
\TEDIT.QUIT	4	(PROCESSP :IN \TEDIT.ACTIVE.WINDOWP)	4

VARIABLE INDEX

TEDIT-CREATION-TIME	1	TEDIT-KILLER-WAIT-TIME ..	1	TEDIT-PROCESS-LIMIT	1	TEDIT-PROCESSES	1
---------------------------	---	---------------------------	---	---------------------------	---	-----------------------	---
