```
(RPAQQ TILED-SEDITCOMS
       [
```

;;; Provides a similar facility for SEdit as the LispUsers package TILEDEDIT provides for DEdit, i.e., instead of prompting the user for regions, generates
;;; successive regions in a circular fashion, eachtime throught the full window loop offsetting the next window by 12,-12. Users can select their
;;; preference through the TILING-ORDER argument to the function TILED.SEDIT.RESET, which must be either NIL (no tiling), T (default tiling order) or
;;; a list of the symbols TL (top-left) TR (top-right) BL (bottom-left) and BR (bottom-right)

```
        ;; User Interface

        (FNS TILED.SEDIT.RESET)

        ;; Support

        (PROP MAKEFILE-ENVIRONMENT TILED-SEDIT)
        (INITVARS (*TiledSEditMargin* 25)
              (*TiledSEditXShift* 15)
              (*TiledSEditYShift* 15)
              (*TiledSEditRegions* NIL))
        (RECORDS TILED.SEDIT.REGION)
        (FNS POST.TILED.SEDIT.GET.WINDOW.REGION POST.TILED.SEDIT.SAVE.WINDOW.REGION TILED.SEDIT.NEW.REGION
              TILED.SEDIT.SWITCHFN)
        (DECLARE%: EVAL@COMPILE DONTEVAL@LOAD DONTCOPY [P (OR (HASDEF 'SEDIT::EDIT-CONTEXT 'RECORDS)
                                                             (EVAL (SYSRECLOOK1 'SEDIT::EDIT-CONTEXT]
              (LOCALVARS . T)
              (GLOBALVARS *TiledSEditXShift* *TiledSEditYShift* *TiledSEditRegions* *TiledSEditRegionWidth*
                      *TiledSEditRegionHeight* *TiledSEditKeepWhenShrunk* *TiledSEditCorners*
                      *TiledSEditNextCornerPtr* *TiledSEditNextTopLeftRegion* *TiledSEditNextBottomLeftRegion*
                      *TiledSEditNextTopRightRegion* *TiledSEditNextBottomRightRegion*))
        (DECLARE%: DONTEVAL@COMPILE DONTEVAL@LOAD DOCOPY (P (TILED.SEDIT.RESET T])
```

;;; Provides a similar facility for SEdit as the LispUsers package TILEDEDIT provides for DEdit, i.e., instead of prompting the user for regions, generates
;;; successive regions in a circular fashion, eachtime throught the full window loop offsetting the next window by 12,-12. Users can select their
;;; preference through the TILING-ORDER argument to the function TILED.SEDIT.RESET, which must be either NIL (no tiling), T (default tiling order) or
;;; a list of the symbols TL (top-left) TR (top-right) BL (bottom-left) and BR (bottom-right)

;; User Interface

```
(DEFINEQ

(TILED.SEDIT.RESET
  [LAMBDA (TILING-ORDER XSHIFT YSHIFT SCREEN)              ; Edited 24-Oct-89 12:09 by Koomen
    [if (NULL TILING-ORDER)
        then ;; Reset the world

              (SETQ *TiledSEditRegions*)
        else ;; Determine new order                        ; BEWARE!!! INFINITE LIST!!!
        [LET [(ORDER (if (EQ TILING-ORDER T)
                         then (LIST :TL :BL :TR :BR)
                         else (for CORNER inside TILING-ORDER collect (SELECTQ CORNER
                                               ((:TL :TOPLEFT :TOP-LEFT :TOP.LEFT)
                                                :TL)
                                               ((:BL :BOTTOMLEFT :BOTTOM-LEFT
                                                 :BOTTOM.LEFT)
                                                :BL)
                                               ((:TR :TOPRIGHT :TOP-RIGHT :TOP.RIGHT)
                                                :TR)
                                               ((:BR :BOTTOMRIGHT :BOTTOM-RIGHT
                                                 :BOTTOM.RIGHT)
                                                :BR)
                                               (ERROR "Unsupported TILING-ORDER spec:"
                                                      CORNER]
              (SETQ TILING-ORDER (COPY ORDER))
              (SETQ *TiledSEditNextCornerPtr* (SETQ *TiledSEditCorners* (NCONC ORDER ORDER]
        ;; Determine starting placements

        (SETQ *TiledSEditXShift* (OR (FIXP XSHIFT)
                                     15))
        (SETQ *TiledSEditYShift* (OR (FIXP YSHIFT)
                                     15))
```

```
          [if (NOT (REGIONP SCREEN))
               then (SETQ SCREEN (LET ((MARGIN (OR (FIXP SCREEN)
                                                    25)))
                                      (CREATEREGION MARGIN MARGIN (IDIFFERENCE SCREENWIDTH MARGIN)
                                                    (IDIFFERENCE SCREENHEIGHT MARGIN]
          (LET* ((WIDTH (LRSH (IDIFFERENCE (fetch (REGION WIDTH) of SCREEN)
                                           (LLSH *TiledSEditXShift* 2))
                              1))
                 (HEIGHT (LRSH (IDIFFERENCE (fetch (REGION HEIGHT) of SCREEN)
                                            (LLSH *TiledSEditYShift* 2))
                               1))
                 (TL-LEFT (fetch (REGION LEFT) of SCREEN))
                 (BL-LEFT TL-LEFT)
                 (BL-BOTTOM (IPLUS *TiledSEditYShift* *TiledSEditYShift* (fetch (REGION BOTTOM) of SCREEN)))
                 (BR-BOTTOM BL-BOTTOM)
                 (TL-BOTTOM (IPLUS BL-BOTTOM HEIGHT *TiledSEditYShift* *TiledSEditYShift*))
                 (TR-BOTTOM TL-BOTTOM)
                 (TR-LEFT (IPLUS TL-LEFT WIDTH *TiledSEditXShift* *TiledSEditXShift*))
                 (BR-LEFT TR-LEFT))
                (SETQ *TiledSEditNextTopLeftRegion* (CREATEREGION TL-LEFT TL-BOTTOM WIDTH HEIGHT))
                (SETQ *TiledSEditNextBottomLeftRegion* (CREATEREGION BL-LEFT BL-BOTTOM WIDTH HEIGHT))
                (SETQ *TiledSEditNextTopRightRegion* (CREATEREGION TR-LEFT TR-BOTTOM WIDTH HEIGHT))
                (SETQ *TiledSEditNextBottomRightRegion* (CREATEREGION BR-LEFT BR-BOTTOM WIDTH HEIGHT)))
          ;; Move currently open SEdit windows (keep relative order), and recompute Tiled SEdit  regions

          (LET ((OLDREGIONS (CAR *TiledSEditRegions*))
                CONTEXT OTHERS)
               (SETQ *TiledSEditRegions* (CONS))
               [for W in (OPENWINDOWS) when (SETQ CONTEXT (WINDOWPROP W 'SEDIT::EDIT-CONTEXT))
                  do (for TSR in OLDREGIONS when (EQ (fetch TSR.CONTEXT of TSR)
                                                     CONTEXT)
                        do (replace TSR.REGION of TSR with W)
                           (RETURN)
                        finally (push OTHERS (CONS W CONTEXT]
               (for TSR in OLDREGIONS when (WINDOWP (fetch TSR.REGION of TSR))
                  do (SHAPEW (fetch TSR.REGION of TSR)
                             (POST.TILED.SEDIT.GET.WINDOW.REGION (fetch TSR.CONTEXT of TSR)
                                    :CREATE)))
               (for PAIR in OTHERS do (SHAPEW (CAR PAIR)
                                              (POST.TILED.SEDIT.GET.WINDOW.REGION (CDR PAIR)
                                                     :CREATE]
        (TILED.SEDIT.SWITCHFN (NULL TILING-ORDER)
               'SEDIT:GET-WINDOW-REGION
               'PRE.TILED.SEDIT.GET.WINDOW.REGION
               'POST.TILED.SEDIT.GET.WINDOW.REGION)
        (TILED.SEDIT.SWITCHFN (NULL TILING-ORDER)
               'SEDIT:SAVE-WINDOW-REGION
               'PRE.TILED.SEDIT.SAVE.WINDOW.REGION
               'POST.TILED.SEDIT.SAVE.WINDOW.REGION)
        TILING-ORDER])
)


;; Support

(PUTPROPS TILED-SEDIT MAKEFILE-ENVIRONMENT (:READTABLE "INTERLISP" :PACKAGE "INTERLISP" :BASE 10))

(RPAQ? *TiledSEditMargin* 25)

(RPAQ? *TiledSEditXShift* 15)

(RPAQ? *TiledSEditYShift* 15)

(RPAQ? *TiledSEditRegions* NIL)

(DECLARE%: EVAL@COMPILE

(RECORD TILED.SEDIT.REGION (TSR.CONTEXT . TSR.REGION))
)

(DEFINEQ

(POST.TILED.SEDIT.GET.WINDOW.REGION
  [LAMBDA (CONTEXT REASON)                                        ; Edited 24-Oct-89 12:17 by Koomen
    (COPY (OR [if (AND SEDIT:KEEP-WINDOW-REGION (EQ REASON :EXPAND))
                  then ;; Make sure it's there (wouldn't be if just enabled KeepWhenShrunk)

                       (for TSR in (CAR *TiledSEditRegions*) when (EQ (fetch TSR.CONTEXT of TSR)
                                                                      CONTEXT)
                          do ;; Don't return the tile, but the actual window region, in case of reshape

                             (RETURN (WINDOWREGION (fetch (SEDIT::EDIT-CONTEXT SEDIT::DISPLAY-WINDOW)
                                                      of CONTEXT]
              (for TSR in (CAR *TiledSEditRegions*) unless (fetch TSR.CONTEXT of TSR)
                 do (replace TSR.CONTEXT of TSR with CONTEXT)
                    (RETURN (fetch TSR.REGION of TSR))
                 finally (SETQ TSR (TILED.SEDIT.NEW.REGION CONTEXT))
```

```
                              (TCONC *TiledSEditRegions* TSR)
                              (RETURN (fetch TSR.REGION of TSR])
```

(**POST.TILED.SEDIT.SAVE.WINDOW.REGION**
```
  [LAMBDA (CONTEXT REASON)                                      ; Edited 24-Oct-89 13:32 by Koomen
     (if (OR (NOT SEDIT:KEEP-WINDOW-REGION)
             (NEQ REASON :SHRINK))
         then (for TSR in (CAR *TiledSEditRegions*) when (EQ (fetch TSR.CONTEXT of TSR)
                                                             CONTEXT)
                 do (RETURN (replace TSR.CONTEXT of TSR with NIL])
```

(**TILED.SEDIT.NEW.REGION**
```
  [LAMBDA (CONTEXT)                                             ; Edited 17-Sep-87 12:51 by Koomen
     (LET* ((NEXTREGION (SELECTQ (pop *TiledSEditNextCornerPtr*)
                                 (:TL *TiledSEditNextTopLeftRegion*)
                                 (:BL *TiledSEditNextBottomLeftRegion*)
                                 (:TR *TiledSEditNextTopRightRegion*)
                                 (:BR *TiledSEditNextBottomRightRegion*)
                                 (SHOULDNT "Bad corner spec!")))
            (THISREGION (COPY NEXTREGION)))
           (replace (REGION LEFT) of NEXTREGION with (IPLUS (fetch (REGION LEFT) of NEXTREGION)
                                                            *TiledSEditXShift*))
           (replace (REGION BOTTOM) of NEXTREGION with (IDIFFERENCE (fetch (REGION BOTTOM) of NEXTREGION)
                                                                    *TiledSEditYShift*))
           (create TILED.SEDIT.REGION
                   TSR.CONTEXT _ CONTEXT
                   TSR.REGION _ THISREGION])
```

(**TILED.SEDIT.SWITCHFN**
```
  [LAMBDA (RESTOREFLG FN SAVEFN REPLFN)                         ; Edited 16-Sep-87 11:18 by Koomen
     (if (NOT (DEFINEDP SAVEFN))
         then (PUTD SAVEFN (GETD FN)))
     (PUTD FN (GETD (if (OR RESTOREFLG (NOT (DEFINEDP REPLFN)))
                        then SAVEFN
                        else REPLFN])
)

(DECLARE%: EVAL@COMPILE DONTEVAL@LOAD DONTCOPY

[OR (HASDEF 'SEDIT::EDIT-CONTEXT 'RECORDS)
    (EVAL (SYSRECLOOK1 'SEDIT::EDIT-CONTEXT]

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *TiledSEditXShift* *TiledSEditYShift* *TiledSEditRegions* *TiledSEditRegionWidth*
        *TiledSEditRegionHeight* *TiledSEditKeepWhenShrunk* *TiledSEditCorners* *TiledSEditNextCornerPtr*
        *TiledSEditNextTopLeftRegion* *TiledSEditNextBottomLeftRegion* *TiledSEditNextTopRightRegion*
        *TiledSEditNextBottomRightRegion*)
)
)

(DECLARE%: DONTEVAL@COMPILE DONTEVAL@LOAD DOCOPY
```

(**TILED.SEDIT.RESET** T)
)

(PUTPROPS **TILED-SEDIT COPYRIGHT** ("Johannes A. G. M. Koomen" 1987 1988 1989))

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX

PROPERTY INDEX