

File created: 19-Oct-87 03:24:45 {PHYLUM}<CAMERON>LISP>TURBO-WINDOWS.;14

changes to: (ADVISE \GETREGIONTRACKWITHBOX DOWINDOWCOM)
(FNS TW.CHECK-DEFAULT-REGION-WANTED TW.CHECK-KEYCHORDS TW.HELP)
(VARS TURBO-WINDOWSCOMS)
(PROPS (TURBO-WINDOWS MAKEFILE-ENVIRONMENT))

previous date: 26-Aug-87 22:48:43 {PHYLUM}<CAMERON>LISP>TURBO-WINDOWS.;12

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1987 by Andrew J Cameron, III and Xerox Corporation. All rights reserved.

(RPAQQ **TURBO-WINDOWSCOMS**

[:: AUTHOR: Andrew J. Cameron, III

:: This file add advice to the function called when the right mouse button is pressed while in a window. It defines a set of accelerators based on other keys which might be depressed when the right mouse button was depressed.

:: Also, a default size for a window can be accessed by using the middle button when originally sizing a window.

::

(PROP MAKEFILE-ENVIRONMENT TURBO-WINDOWS)

:: Would like to know the number of pixels per inch here

(INITVARS (GETREGIONDEFAULT (CONS (ITIMES 7 72)
(ITIMES 9 72)))

(TW.NO-FLASH-CLOSE)
(TW.DONT-GROW-SNAP-BORDER)
(TW.SNAP-HERE)

(SPECVARS GETREGIONDEFAULT TW.NO-FLASH-CLOSE TW.DONT-GROW-SNAP-BORDER TW.SNAP-HERE)

:: Need access to source-code for WINDOWS because we need the definition of record SCREEN.

(DECLARE: DONTVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
WINDOW))

(FNS TW.CHECK-DEFAULT-REGION-WANTED TW.HELP SNAPW-WINDOW TW.CHECK-KEYCHORDS)
(ADVISE DOWINDOWCOM \GETREGIONTRACKWITHBOX)

:: This adds SNAP-WINDOW as a slide-out from SNAP on IL:WindowMenuCommands

(P (LET ((foo (SASSOC (QUOTE Snap)
WindowMenuCommands)))
(if foo then [NCONC foo (QUOTE ((SUBITEMS ("Snap Window" (QUOTE SNAPW-WINDOW)
"Saves a snapshot of this window"))
(SUBITEMS ("Snap Window" (QUOTE SNAPW-WINDOW)
"Saves a snapshot of this window"]
(SETQ WindowMenu])

:: AUTHOR: Andrew J. Cameron, III

:: This file add advice to the function called when the right mouse button is pressed while in a window. It defines a set of accelerators based on other keys which might be depressed when the right mouse button was depressed.

:: Also, a default size for a window can be accessed by using the middle button when originally sizing a window.

::

(PUTPROPS **TURBO-WINDOWS MAKEFILE-ENVIRONMENT** (PACKAGE "INTERLISP" READTABLE "OLD-INTERLISP-FILE"))

:: Would like to know the number of pixels per inch here

(RPAQ? **GETREGIONDEFAULT** (CONS (ITIMES 7 72)
(ITIMES 9 72)))

(RPAQ? **TW.NO-FLASH-CLOSE**)

(RPAQ? **TW.DONT-GROW-SNAP-BORDER**)

(RPAQ? **TW.SNAP-HERE**)

(DECLARE: DOEVAL@COMPILE DONTCOPY

(SPECVARS GETREGIONDEFAULT TW.NO-FLASH-CLOSE TW.DONT-GROW-SNAP-BORDER TW.SNAP-HERE)
)

:: Need access to source-code for WINDOWS because we need the definition of record SCREEN.

(DECLARE: DONTVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
WINDOW)

)

(DEFINEQ

(TW.CHECK-DEFAULT-REGION-WANTED

[LAMBDA NIL

; Edited 19-Oct-87 03:21 by andyiii

:: Provides a default region if the middle button is depressed during the initial sizing of a window.

```
(DECLARE (SPECVARS GETREGIONDEFAULT BASEX BASEY OPPX OPPY))
(if (AND (BOUNDP (QUOTE GETREGIONDEFAULT))
        (POSITIONP GETREGIONDEFAULT)
        (MOUSESTATE MIDDLE))
    then (LET* ((x (fetch XCOORD of GETREGIONDEFAULT))
               (y (fetch YCOORD of GETREGIONDEFAULT))
               (maxx (IPLUS BASEX x))
               (miny (IDIFFERENCE BASEY y)))
           (if (GREATERP maxx SCREENWIDTH)
               then (SETQ BASEX SCREENWIDTH)
                  (SETQ OPPX (IDIFFERENCE SCREENWIDTH x))
               else (SETQ OPPX maxx))
           (if (LESSP miny 0)
               then (SETQ BASEY 0)
                  (SETQ OPPY y)
               else (SETQ OPPY miny]))
```

(TW.HELP

[LAMBDA NIL

; Edited 19-Oct-87 02:17 by andyiii

:: Print out a cribsheet to remind users of what to do.

```
(PRINTOUT T "TurboWindow CribSheet [S=Shift C=Control M=Meta]" T T)
(PRINTOUT T "-ACTION- -CHORD- -OTHER WAY- -BACKGROUND-" T)
(PRINTOUT T " Copy S <COPY> Snap" T)
(PRINTOUT T " Delete C <DEL>* SlamWindows" T)
(PRINTOUT T " Move SC <MOVE>" T)
(PRINTOUT T " Shape M <SAME>* PSW" T)
(PRINTOUT T " Shrink MC <META-SAME>* Idle" T)
(PRINTOUT T " Bury SMC <FIND>* T)
(PRINTOUT T " Redisplay MS <AGAIN>* T)
(PRINTOUT T " Clear <RSHIFT> <META-AGAIN>* Idle" T)
(PRINTOUT T " Hardcopy <META-COPY>" T)
(PRINTOUT T " Inspect <META-HELP>* PSW" T)
(PRINTOUT T " PopShape <UNDO>* T)
(PRINTOUT T " This Help <HELP> Help" T)
(PRINTOUT T "[* = Presssing this key may have side-effects]"])
```

(SNAPW-WINDOW

[LAMBDA (WINDOW ANDATTACHEDWINDOWSTOO?)

; Edited 22-Aug-87 00:24 by andyiii

:: makes a new window which is a copy of the bits in the WINDOW passed in.

```
(PROG (SCREENREGION SCREEN REGION NEWWINDOW BorderGrowth)
      (SETQ BorderGrowth (if (AND (QUOTE TW.DONT-GROW-SNAP-BORDER))
                             TW.DONT-GROW-SNAP-BORDER)
                             then 0
                             else WBorder))
  (TOTOPW WINDOW)
  (SETQ SCREEN (WINDOWPROP WINDOW (QUOTE SCREEN)))
  [SETQ REGION (if ANDATTACHEDWINDOWSTOO?
                  then (WINDOWREGION WINDOW)
                  else (WINDOWPROP WINDOW (QUOTE REGION))]
  (SETQ NEWWINDOW (CREATEW (create SCREENREGION
                                   SCREEN _ SCREEN
                                   REGION _ (GROW/REGION REGION BorderGrowth))
                           NIL NIL T))
  ; keep it closed so it doesn't cover any of the bits it is to copy.
  ; put existing screen bits from SAVE.
  (BITBLT (fetch (SCREEN SCDESTINATION) of SCREEN)
          (fetch (REGION LEFT) of REGION)
          (fetch (REGION BOTTOM) of REGION)
          (fetch (WINDOW SAVE) of NEWWINDOW)
          BorderGrowth BorderGrowth (fetch (REGION WIDTH) of REGION)
          (fetch (REGION HEIGHT) of REGION)
          (QUOTE INPUT)
          (QUOTE REPLACE))
  (OPENW NEWWINDOW)
  (if (AND (BOUNDP (QUOTE TW.SNAP-HERE))
          TW.SNAP-HERE)
      else (MOVEW NEWWINDOW))
  (RETURN NEWWINDOW])
```

(TW.CHECK-KEYCHORDS

[LAMBDA (WINDOW)

; Edited 19-Oct-87 02:18 by andyiii

```
(LET [(THE-WINDOW (OR (AND (BOUNDP (QUOTE WINDOW))
                          WINDOW)
                     (WHICHW]
      [SETQ THE-WINDOW (if THE-WINDOW
                          then (if (KEYDOWNP (QUOTE SPACE))
                                  then THE-WINDOW
```

else (MAINWINDOW THE-WINDOW T]

:: (printout t "Window " the-window t)

::

:: I want to use (shiftdownp 'meta) but it seems I must use (keydownp 'keyboard)

:: This happens when the KEYBOARD key is not defined to be the META key.

(COND

((AND (SHIFTDOWNP (QUOTE 1SHIFT))
(SHIFTDOWNP (QUOTE 2SHIFT)))
(PRINTOUT T "How did you do that?" t)
[GREMLINS (AND (SHIFTDOWNP (QUOTE CTRL))
(SHIFTDOWNP (QUOTE META))
"RETURN")
([OR (SHIFTDOWNP (QUOTE 2SHIFT))
(AND (KEYDOWNP (QUOTE AGAIN))
(SHIFTDOWNP (QUOTE META))

::: CLEAR (must come before tests for SHIFT

(if THE-WINDOW
then (CLEARW THE-WINDOW)
else (IDLE))
"RETURN")
((KEYDOWNP (QUOTE OPEN))

; the HELP key

::: INSPECT

(if (SHIFTDOWNP (QUOTE META))
then (if THE-WINDOW
then (INSPECT THE-WINDOW)
else (PROCESS.STATUS.WINDOW))
else (TW.HELP))
"RETURN")
([OR (AND (KEYDOWNP (QUOTE SAME))
(NOT (SHIFTDOWNP (QUOTE SHIFT)))
(NOT (SHIFTDOWNP (QUOTE CTRL)))
(SHIFTDOWNP (QUOTE META)))
(AND (NOT (KEYDOWNP (QUOTE SAME)))
(NOT (SHIFTDOWNP (QUOTE SHIFT)))
(SHIFTDOWNP (QUOTE CTRL))
(SHIFTDOWNP (QUOTE META))

::: SHRINK

(if THE-WINDOW
then (SHRINKW THE-WINDOW)
else (IDLE))
"RETURN")
([OR (KEYDOWNP (QUOTE MOVE))
(AND (SHIFTDOWNP (QUOTE SHIFT))
(SHIFTDOWNP (QUOTE CTRL))
(NOT (SHIFTDOWNP (QUOTE META))

::: MOVE

(MOVEW THE-WINDOW)
"RETURN")
([OR (AND (KEYDOWNP (QUOTE SAME))
(NOT (SHIFTDOWNP (QUOTE SHIFT)))
(NOT (SHIFTDOWNP (QUOTE CTRL)))
(NOT (SHIFTDOWNP (QUOTE META))
(AND (NOT (KEYDOWNP (QUOTE SAME)))
(NOT (SHIFTDOWNP (QUOTE SHIFT)))
(NOT (SHIFTDOWNP (QUOTE CTRL)))
(SHIFTDOWNP (QUOTE META))

::: SHAPE

(if THE-WINDOW
then (if (GETD (QUOTE SHAPEW.AND.SAVE))
then (SHAPEW.AND.SAVE THE-WINDOW)
else (SHAPEW THE-WINDOW))
else (PROCESS.STATUS.WINDOW))
"RETURN")
([OR (KEYDOWNP (QUOTE COPY))
(AND (SHIFTDOWNP (QUOTE SHIFT))
(NOT (SHIFTDOWNP (QUOTE CTRL)))
(NOT (SHIFTDOWNP (QUOTE META))

::: COPY (snap of hardcopy)

(if (AND (KEYDOWNP (QUOTE COPY))
(SHIFTDOWNP (QUOTE SHIFT)))
then (LET [(FN (MENU (create MENU
ITEMS _ (QUOTE ("To default printer" (QUOTE HARDCOPYIMAGEW)
"Prints a window using its HARDCOPYFN."))

```

("To a file" (QUOTE HARDCOPYIMAGEW.TOFILE)
  "Puts image on a file; prompts for filename
  and format")
("To another printer" (QUOTE
  HARDCOPYIMAGEW.TOPRINTER
  )
  "Sends image to a printer of your choosing"))

```

```

)
CHANGEOFFSETFLG _ (QUOTE Y)
MENUOFFSET _ (create POSITION
  XCOORD _ -1
  YCOORD _ 0)
WHENHELDFN _ (FUNCTION PFPROMPT3)
WHENUNHELDFN _ (FUNCTION CLRPPROMPT)
CENTERFLG _ T]
  (if FN
    then (APPLY* FN THE-WINDOW))
  else (SNAPW-WINDOW (TOTOPW THE-WINDOW)
    (NOT (KEYDOWNP (QUOTE SPACE)
      "RETURN")
      ([OR (KEYDOWNP (QUOTE AGAIN))
        (AND (SHIFTDOWNP (QUOTE SHIFT))
          (NOT (SHIFTDOWNP (QUOTE CTRL)))
          (SHIFTDOWNP (QUOTE META)

```

::: REDISPLAY

```

(if THE-WINDOW
  then (REDISPLAYW THE-WINDOW))
"RETURN"
([OR (KEYDOWNP (QUOTE FIND))
  (AND (SHIFTDOWNP (QUOTE SHIFT))
    (SHIFTDOWNP (QUOTE CTRL))
    (SHIFTDOWNP (QUOTE META)

```

::: BURY

```

(if THE-WINDOW
  then (BURYW THE-WINDOW))
"RETURN"
([OR (KEYDOWNP (QUOTE DELETE))
  (AND (NOT (SHIFTDOWNP (QUOTE SHIFT)))
    (SHIFTDOWNP (QUOTE CTRL))
    (NOT (SHIFTDOWNP (QUOTE META)

```

::: CLOSE

```

(if THE-WINDOW
  then (if (AND (BOUNDP (QUOTE TW.NO-FLASH-CLOSE))
    TW.NO-FLASH-CLOSE)
    else (FLASHWINDOW THE-WINDOW)) ; show the user the window about to be closed
    (while (KEYDOWNP (QUOTE RIGHT))) ; only close it if they have not release teh delete key nor moved
    (GETMOUSESTATE) ; out of the window
    (if (AND (OR (SHIFTDOWNP (QUOTE CTRL))
      (KEYDOWNP (QUOTE DELETE)))
      (INSIDEP (WINDOWPROP THE-WINDOW (QUOTE REGION))
        LASTMOUSEX LASTMOUSEY))
      then (CLOSEW THE-WINDOW)
        "RETURN"
    else ; the background was selected
      (if (GETD (QUOTE CLOSE.WINDOWS.IN.REGION))
        then (PROMPTPRINT "Specify the Slamming region")
          (CLOSE.WINDOWS.IN.REGION)
        else (PROMPTPRINT "Window slamming is not available - You need to load WDHACKS"))
        "RETURN"))
    ([OR (KEYDOWNP (QUOTE UNDO))
      (AND (KEYDOWNP (QUOTE SAME))
        (SHIFTDOWNP (QUOTE SHIFT)

```

::: POP-SHAPE

```

(if THE-WINDOW
  then (if (GETD (QUOTE SHAPEW.POP))
    THEN (SHAPEW.POP THE-WINDOW)
    ELSE (PROMPTPRINT "Window shape-popping is not available - You need to load WDHACKS")))
"RETURN"]
)

```

```

[XCL REINSTALL-ADVICE (QUOTE DOWINDOWCOM)
  BEFORE
  (QUOTE (( FIRST (if (EQUAL (TW.CHECK-KEYCHORDS WINDOW)
    "RETURN")
    then
    (RETURN]

```

```
{MEDLEY}<lispusers>TURBO-WINDOWS.;1

[XCL REINSTALL-ADVICE (QUOTE \GETREGIONTRACKWITHBOX)
  AFTER
  (QUOTE (( FIRST (TW.CHECK-DEFAULT-REGION-WANTED]

(READWISE DOWINDOWCOM \GETREGIONTRACKWITHBOX)

;; This adds SNAP-WINDOW as a slide-out from SNAP on IL:WindowMenuCommands

(LET ((foo (SASSOC (QUOTE Snap)
                  WindowMenuCommands)))
  (if foo
      then [NCONC foo (QUOTE ((SUBITEMS ("Snap Window" (QUOTE SNAPW-WINDOW)
                                          "Saves a snapshot of this window"))
                                      (SUBITEMS ("Snap Window" (QUOTE SNAPW-WINDOW)
                                          "Saves a snapshot of this window"]
                                          (SETQ WindowMenu)))
  (PUTPROPS TURBO-WINDOWS COPYRIGHT ("Andrew J Cameron, III and Xerox Corporation" 1987))
```

FUNCTION INDEX

SNAPW-WINDOW2 TW.CHECK-KEYCHORDS2
TW.CHECK-DEFAULT-REGION-WANTED2 TW.HELP2

VARIABLE INDEX

GETREGIONDEFAULT1 TW.DONT-GROW-SNAP-BORDER 1 TW.NO-FLASH-CLOSE1 TW.SNAP-HERE1

ADVICE INDEX

DOWINDOWCOM4 \GETREGIONTRACKWITHBOX ..5

PROPERTY INDEX

TURBO-WINDOWS1
