

File created: 8-Jul-86 17:40:07 {ERIS}<LISPUSERS>LISPCORE>PAGEHOLD.;3

changes to: (VARS PAGEHOLDCOMS)  
(MACROS CTRLREALLYDOWN?)  
(FNS END.OF.PAGE.HOLD \EOP.DO.BUTTON)

previous date: 15-Apr-86 11:21:20 {ERIS}<LISPUSERS>LISPCORE>PAGEHOLD.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(\* \* Copyright (c) 1982, 1983, 1984, 1985, 1986 by Xerox Corporation.  
All rights reserved.)

```
(RPAQQ PAGEHOLDCOMS
((COMS (* "Parameters adjustable by user.")
[INITVARS (PAGE.WAIT.SECONDS 20)
(PAGE.WAIT.ACTIVITY (QUOTE WINKING))
(PAGE.WAIT.IGNORETYPEAHEAD NIL)
(PAGE.WAIT.FONT (FONTCREATE (QUOTE HELVETICA)
12))
(PAGE.WAIT.HOLDMSG (QUOTE (" -- SHIFT to hold typeout -- " 198)))
(PAGE.WAIT.RELEASEMSG (QUOTE (" -- Release SHIFT for more -- " 215)))
(PAGE.WAIT.STOPMSG (QUOTE (" -- Scrolling Stopped -- " 169])
(GLOBALVARS PAGE.WAIT.SECONDS PAGE.WAIT.ACTIVITY PAGE.WAIT.IGNORETYPEAHEAD PAGE.WAIT.FONT
PAGE.WAIT.HOLDMSG PAGE.WAIT.RELEASEMSG PAGE.WAIT.STOPMSG))
(DECLARE: DONTCOPY (MACROS TYPEAHEADP CTRLREALLYDOWN?)
(RECORDS PAGEHOLDBUTTON PAGEHOLDMSG))
[VARS (HoldingButtonMenu NIL)
(HoldButtonBottomLine (FONTDESCENT PAGE.WAIT.FONT))
(TitleBarHeight (FONTHEIGHT (DSPFONT NIL WindowTitleDisplayStream]
(GLOBALVARS HoldingButtonMenu HoldButtonBottomLine TitleBarHeight)
(FNS END.OF.PAGE.HOLD \EOP.DO.BUTTON \PageHold.printMessage \PageHold.buttonEventFn \PageHold.doMenu)
(COMS (* "Items related to the PAGEHOLDBUTTON resource")
(DECLARE: DONTCOPY (RESOURCES PAGEHOLDBUTTON))
(INITRESOURCES PAGEHOLDBUTTON)
(FNS MakePageHoldButton \PageHold.GET)
(GLOBALVARS \PAGEHOLDBUTTONS))
[DECLARE: DONTVAL@LOAD DOCOPY (P (MOVED? (QUOTE PAGEFULLFN)
(QUOTE OLDPAGEFULLFN)
NIL T)
(/MOVED (QUOTE END.OF.PAGE.HOLD)
(QUOTE PAGEFULLFN]
(LOCALVARS . T)))
```

:: Parameters adjustable by user.

```
(RPAQ? PAGE.WAIT.SECONDS 20)
(RPAQ? PAGE.WAIT.ACTIVITY (QUOTE WINKING))
(RPAQ? PAGE.WAIT.IGNORETYPEAHEAD NIL)
(RPAQ? PAGE.WAIT.FONT (FONTCREATE (QUOTE HELVETICA)
12))
(RPAQ? PAGE.WAIT.HOLDMSG (QUOTE (" -- SHIFT to hold typeout -- " 198)))
(RPAQ? PAGE.WAIT.RELEASEMSG (QUOTE (" -- Release SHIFT for more -- " 215)))
(RPAQ? PAGE.WAIT.STOPMSG (QUOTE (" -- Scrolling Stopped -- " 169)))
(DECLARE: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS PAGE.WAIT.SECONDS PAGE.WAIT.ACTIVITY PAGE.WAIT.IGNORETYPEAHEAD PAGE.WAIT.FONT PAGE.WAIT.HOLDMSG
PAGE.WAIT.RELEASEMSG PAGE.WAIT.STOPMSG)
)
(DECLARE: DONTCOPY
(DECLARE: EVAL@COMPILE
(PUTPROPS TYPEAHEADP MACRO (NIL (AND (NOT PAGE.WAIT.IGNORETYPEAHEAD)
(READP T))))))
(PUTPROPS CTRLREALLYDOWN? MACRO [(N)
(AND (SHIFTDOWNP (QUOTE CTRL))
(PROGN
```

(\* Consider it a spazz if he didn't keep the CTRL key down for at least N milliseconds)

```
(DISMISS N NIL T)
(SHIFTDOWNP (QUOTE CTRL])
```

```

)

(DECLARE: EVAL@COMPILE

(RECORD PAGEHOLDBUTTON (TIMERS ACTIVITY BUTTONIMAGE PAGEHOLDBUTTONWIDTH))

(RECORD PAGEHOLDMSG (MSG WIDTH))
)
)

(RPAQQ HoldingButtonMenu NIL)

(RPAQ HoldButtonBottomLine (FONTDESCENT PAGE.WAIT.FONT))

(RPAQ TitleBarHeight (FONTHEIGHT (DSPFONT NIL WindowTitleDisplayStream)))

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS HoldingButtonMenu HoldButtonBottomLine TitleBarHeight)
)

(DEFINEQ

(END.OF.PAGE.HOLD
  [LAMBDA (STREAM)
    (LET* ([WINDOW (WFROMDS (COND
      ((NULL STREAM)
        (TTYDISPLAYSTREAM))
      (T (\DTEST STREAM (QUOTE STREAM)
        (WAIT.SECS (OR (WINDOWPROP WINDOW (QUOTE PAGE.WAIT.SECONDS))
          PAGE.WAIT.SECONDS)))
      (COND
        ((OR (NULL WAIT.SECS)
          (AND (NOT (FIXP WAIT.SECS))
            (NEQ WAIT.SECS (QUOTE STOP)))
          (EQ (DSPSCROLL NIL WINDOW)
            (QUOTE OFF))))
        (* If we're losing because of an invalid value in PAGE.WAIT.SECONDS then try to fix it up.)

        (COND
          ((NULL (WINDOWPROP WINDOW (QUOTE PAGE.WAIT.SECONDS)))
            (SETQ PAGE.WAIT.SECONDS 0)))
        T)
        ((AND (NOT (SHIFTDOWNP (QUOTE SHIFT)))
          (NEQ WAIT.SECS (QUOTE STOP))
          (OR (ILESSP WAIT.SECS 1)
            (CTRLREALLYDOWN? 125)
            (TYPEAHEADP)))
          (* Immediate release case)

          T)
        (T (\EOP.DO.BUTTON STREAM WINDOW WAIT.SECS]))
    ]
    (* bvm: " 8-Jul-86 12:41")

(EOP.DO.BUTTON
  [LAMBDA (STREAM WINDOW WAIT.SECS)
    (RESETLST
      (* A RESETLST so that the button can be forced down, regardless of how things got exited)

      (LET* ((CURRENTBUTTON (GETRESOURCE PAGEHOLDBUTTON (OR (WINDOWPROP WINDOW (QUOTE PAGE.WAIT.ACTIVITY))
        PAGE.WAIT.ACTIVITY)))
        (BORDERSIZE (OR (FIXP (WINDOWPROP WINDOW (QUOTE BORDER))
          0))
        (REG (WINDOWPROP WINDOW (QUOTE REGION)))
        (LEFT (IDIFFERENCE (IDIFFERENCE (fetch (REGION PRIGHT) of REG)
          BORDERSIZE)
          (fetch PAGEHOLDBUTTONWIDTH of CURRENTBUTTON)))
        (BOTTOM (IDIFFERENCE (fetch (REGION PTOP) of REG)
          (IPLUS TitleBarHeight BORDERSIZE)))
        (BUTTON.WINDOW (fetch BUTTONIMAGE of CURRENTBUTTON)))
        [COND
          ((IGREATERP LEFT (IDIFFERENCE SCREENWIDTH 25))
            (* If the right edge of the window is almost off the screen then put the "button" on the left side.)

            (SETQ LEFT (IPLUS BORDERSIZE (fetch (REGION LEFT) of REG)
              [COND
                ((IGREATERP BOTTOM (IDIFFERENCE SCREENHEIGHT 12))
                  (* If the top of the window is almost off the screen then put the "button" on the bottom.)

                  (SETQ BOTTOM (IPLUS BORDERSIZE (IDIFFERENCE (fetch (REGION BOTTOM) of REG)
                    (WINDOWPROP (fetch BUTTONIMAGE of CURRENTBUTTON)
                      (QUOTE HEIGHT))
                    (PROG ((INFINITY MAX.FIXP)
                      (BUTTON.WINDOW (fetch BUTTONIMAGE of CURRENTBUTTON))
                      (ACTIVITY (fetch ACTIVITY of CURRENTBUTTON))

```

```

(FIRSTIMEP T)
(MESSAGESTATE 1)
FLASHINTERVAL MESSAGESTATEINIT STOPFLG INDEFINITEHOLD HOLDPROP WAITTIMER FLASHTIMER
MENSIGNAL TIMERSLST)
(WINDOWPROP BUTTON.WINDOW (QUOTE END.OF.PAGE.HOLD)
  T)
(MOVEW BUTTON.WINDOW LEFT BOTTOM)
(DSPRESET BUTTON.WINDOW)
[RESETSAVE (PROGN BUTTON.WINDOW
  (QUOTE (AND (WINDOWP OLDVALUE)
    (CLOSEW OLDVALUE]
  (OPENW BUTTON.WINDOW)
INITIALIZETIMERS
(SETQ TIMERSLST (fetch TIMERS of CURRENTBUTTON))
(SETQ FLASHTIMER (SETUPTIMER 0 (pop TIMERSLST)
  (QUOTE TICKS)))
[SETQ FLASHINTERVAL (COND
  [(EQ ACTIVITY (QUOTE FLASHING))
    (SETQ MESSAGESTATEINIT 4)
    (SELECTC \MACHINETYPE
      (\DANDELION (CONSTANT (TIMES 1250 \DLION.RCLKMILLISECOND)))
      (CONSTANT (TIMES 1250 \ALTO.RCLKMILLISECOND]
    (T (* So it's WINKING or NIL)
      (SETQ MESSAGESTATEINIT 2)
      (SELECTC \MACHINETYPE
        (\DANDELION (CONSTANT (TIMES 750 \DLION.RCLKMILLISECOND)))
        (CONSTANT (TIMES 750 \ALTO.RCLKMILLISECOND]
  ]
[SETQ WAITTIMER (COND
  ((EQ WAIT.SECS (QUOTE STOP))
    (* Initialization done in case a menu selection changes state)
    (SETQ FIRSTIMEP (SETQ STOPFLG T))
    (SETQ MESSAGESTATE (SETQ MESSAGESTATEINIT 1))
    NIL)
  (T (SETQ STOPFLG)
    (SETUPTIMER WAIT.SECS (pop TIMERSLST)
      (QUOTE SECONDS]
  (AND INDEFINITEHOLD (BLOCK 375))
LOOP
[COND
  (FLASHTIMER (COND
    ((AND (SHIFTDOWNP (QUOTE SHIFT))
      (OR (NULL INDEFINITEHOLD)
        (TIMEREXPIRED? INDEFINITEHOLD (QUOTE SECONDS)))
      (NOT STOPFLG))
      (* Lock in on holding message while SHIFT is down)
      (DSPRESET BUTTON.WINDOW)
      (\PageHold.printMessage CURRENTBUTTON T PAGE.WAIT.RELEASEMSG)
      (SETQ WAITTIMER (SETQ FLASHTIMER)))
    ((TIMEREXPIRED? FLASHTIMER (QUOTE TICKS))
      (COND
        [(OR FIRSTIMEP (AND ACTIVITY (NOT STOPFLG)))
          (SETQ FIRSTIMEP)
          (COND
            ((ILEQ (add MESSAGESTATE -1)
              0)
              (\PageHold.printMessage CURRENTBUTTON T (COND
                (STOPFLG
                  PAGE.WAIT.STOPMSG)
                (T PAGE.WAIT.RELEASEMSG)
              )
              (SETQ MESSAGESTATE MESSAGESTATEINIT))
              (INDEFINITEHOLD (DSPRESET BUTTON.WINDOW))
              (T (SELECTQ ACTIVITY
                (WINKING (\PageHold.printMessage CURRENTBUTTON NIL
                  PAGE.WAIT.HOLDMSG))
                (FLASHING [COND
                  ((ODDP MESSAGESTATE)
                    (DSPRESET BUTTON.WINDOW))
                  (T (\PageHold.printMessage CURRENTBUTTON NIL
                    (COND
                      ((IGEQ MESSAGESTATE 2)
                        PAGE.WAIT.HOLDMSG)
                      (T PAGE.WAIT.RELEASEMSG])
                    )
                  NIL]
                )
              (* Make sure the button continues to be visible, even when there
              is no activity)
              (TOTOPW BUTTON.WINDOW)))
              (SETQ FLASHTIMER (SETUPTIMER FLASHINTERVAL FLASHTIMER (QUOTE TICKS]
            )
          (BLOCK)
          [SELECTQ (SETQ HOLDPROP (WINDOWPROP BUTTON.WINDOW (QUOTE END.OF.PAGE.HOLD)))
            ((NIL END.OF.PAGE.HOLD)
              (* Release by simple LEFT mousing)
              (RETURN T))
            (MENU)
              (* Aha, some intervention via MENU so first restore the
              windowprop to the "waiting" state.)
              (WINDOWPROP BUTTON.WINDOW (QUOTE END.OF.PAGE.HOLD)
                T)
              (SELECTQ (SETQ MENSIGNAL (\PageHold.doMenu))
                (END.OF.PAGE.HOLD

```

```

        (RETURN T))
    (PAGE.WAIT.SECONDS
     (FRESHLINE PROMPTWINDOW)
     (SETQ WAIT.SEC (MKATOM (PROMPTFORWORD "Default holding timelimit for this
                                         window = " WAIT.SEC NIL PROMPTWINDOW NIL
                                         30)))
     (COND
      ((NUMBERP WAIT.SEC)
       (SETQ WAIT.SEC (FIX WAIT.SEC)))
      ((EQ WAIT.SEC (QUOTE STOP)))
      (T (SETQ WAIT.SEC)))
     (WINDOWPROP WINDOW (QUOTE PAGE.WAIT.SECONDS)
                  WAIT.SEC)
     (SELECTQ WAIT.SEC
      (0 (RETURN T))
      (NIL (SETQ WAIT.SEC PAGE.WAIT.SECONDS)
            NIL)
      (GO INITIALIZETIMERS)))
    ((0 INFINITY DEFAULT STOP)
     (SETQ WAIT.SEC (SELECTQ MENUSIGNAL
                             (0 0)
                             (INFINITY INFINITY)
                             (DEFAULT PAGE.WAIT.SECONDS)
                             (STOP (QUOTE STOP))
                             NIL))
     (FLASHWINDOW WINDOW)
     (PROMPTPRINT "Setting Default timelimit for this window to " MENUSIGNAL)
     (WINDOWPROP WINDOW (QUOTE PAGE.WAIT.SECONDS)
                  (AND (NEQ MENUSIGNAL (QUOTE DEFAULT))
                       WAIT.SEC))
     (COND
      ((EQ 0 WAIT.SEC)
       (RETURN))
      (T (GO INITIALIZETIMERS))))
    (T (GO SETUPINDEFINITEHOLD))
    (PAGE.WAIT.STOPMSG
     (SETQ WAIT.SEC (QUOTE STOP))
     (GO INITIALIZETIMERS))
    NIL)
  (COND
   ((TYPEAHEADP) (* Flush the character he typed to "release")
    (\GETKEY)
    (RETURN))
   ((AND (NOT STOPFLG)
        (CTRLREALLYDOWN? 125))
    (COND
     ((NOT (SHIFTDOWNP (QUOTE SHIFT)))
      (* If only the CTRL key is down, then immediate release.)
      (RETURN))
     ((OR (NULL INDEFINITEHOLD)
          (TIMEREXPIRED? INDEFINITEHOLD (QUOTE SECONDS)))
      (GO SETUPINDEFINITEHOLD))
    ))
  (COND
   ((AND WAITTIMER (TIMEREXPIRED? WAITTIMER (QUOTE SECONDS)))
    (SETQ WAITTIMER))
  (COND
   ((AND (NOT STOPFLG)
        (NULL WAITTIMER)
        (NOT (SHIFTDOWNP (QUOTE SHIFT))
         (* Basic return from timeout, where no holding action is present)
         (RETURN T)))
    (GO LOOP)
    SETUPINDEFINITEHOLD
    (SETQ WAIT.SEC INFINITY)
    (SETQ INDEFINITEHOLD (SETUTIMER 5 (OR INDEFINITEHOLD (pop TIMERSLST))
                                     (QUOTE SECONDS)))
    (GO INITIALIZETIMERS))
  (FREERESOURCE PAGEHOLDBUTTON CURRENTBUTTON)
  T))]]

```

## (\PageHold.printMessage

(\* JonL " 1-Dec-84 17:03")

```

[LAMBDA (PAGEHOLDBUTTON BOTTOMP MSG)
  (DECLARE (GLOBALVARS HoldButtonBottomLine))
  (LET ((BUTTON.WINDOW (fetch BUTTONIMAGE of PAGEHOLDBUTTON))
        (DSPRESET BUTTON.WINDOW)
        (if BOTTOMP
         then (DSPYPOSITION HoldButtonBottomLine BUTTON.WINDOW)
         (DSPXPOSITION (LRSH (DIFFERENCE (fetch PAGEHOLDBUTTONWIDTH of PAGEHOLDBUTTON)
                                         (fetch (PAGEHOLDMSG WIDTH) of MSG))
                       1)
          BUTTON.WINDOW)
        (PRIN3 (fetch (PAGEHOLDMSG MSG) of MSG)
                BUTTON.WINDOW])

```

## (\PageHold.buttonEventFn

(\* JonL " 5-Oct-84 23:54")

[LAMBDA (W)

```

(WINDOWPROP W (QUOTE END.OF.PAGE.HOLD)
  (if (LASTMOUSESTATE MIDDLE)
    then (QUOTE MENU)
    elseif (LASTMOUSESTATE LEFT)
    then (QUOTE END.OF.PAGE.HOLD)
    else))

```

## (\PageHold.doMenu

```

[LAMBDA NIL
  (MENU (OR HoldingButtonMenu (SETQ HoldingButtonMenu (create MENU
    (* JonL " 6-Oct-84 18:28")
    ITEMS _ (QUOTE ("set Window Wait to read-in"
      (QUOTE PAGE.WAIT.SECONDS)
      "Window gets new PAGE.WAIT.SECONDS
      property from type-in.")
      ("set Window Wait to infinity"
      (QUOTE INFINITY)
      "Set Window's PAGE.WAIT.SECONDS
      prop to infinity")
      ("set Window Wait to 0" 0 "Set
      Window's PAGE.WAIT.SECONDS
      prop to 0")
      ("use default Wait value"
      (QUOTE DEFAULT)
      "Remove Window's PAGE.WAIT.SECONDS
      property")
      ("set Window Wait to 'stop'"
      (QUOTE T)
      "Set Window's PAGE.WAIT.SECONDS
      prop for 'stopping' mode")
      ("Keep this hold indefinitely"
      (QUOTE T)
      "Go into indefinite hold mode")
      ("simple 'stop' now" (QUOTE
        HoldMessage.stop
        )
        "Puts current hold into
        'stopped' state")
      ("Release this hold!"
      (QUOTE END.OF.PAGE.HOLD)
      "Simple release from holding"))))
    MENUBORDERSIZE _ 1
    TITLE _ "Window Wait Options"]])
)

```

;; Items related to the PAGEHOLDBUTTON resource

```

(DECLARE: DONTCOPY

(DECLARE: EVAL@COMPILE

[PUTDEF (QUOTE PAGEHOLDBUTTON)
  (QUOTE RESOURCES)
  (QUOTE (NEW (MakePageHoldButton . ARGS)
    FREE
    (RPLACA (OR (find L on (CDR \PAGEHOLDBUTTONS)
      suchthat
      (NULL (CAR L)))
      (LAST (NCONC1 \PAGEHOLDBUTTONS NIL))))
    (PROG1 . ARGS))
    GET
    (\PageHold.GET . ARGS)
    INIT
    (/SETTOPVAL (QUOTE \PAGEHOLDBUTTONS)
      (LIST NIL])
  )
)

```

```

(/SETTOPVAL (QUOTE \PAGEHOLDBUTTONS)
  (LIST NIL))

```

(DEFINEQ

## (\MakePageHoldButton

```

[LAMBDA (ACTIVITY)
  (PROG ((BORDERSIZE (SELECTQ (OR ACTIVITY PAGE.WAIT.ACTIVITY)
    (* JonL " 1-Dec-84 17:04")
    ((WINKING)
      8)
    ((FLASHING)
      2)
    ((NIL)
      0)
    (PROGN (if ACTIVITY
      then (\ILLEGAL.ARG ACTIVITY))
      (* Patch up the global variable)
      (SETQ PAGE.WAIT.ACTIVITY)

```

```

0)))
(MOREWIDTH (STRINGWIDTH (fetch (PAGEHOLDMSG MSG) of PAGE.WAIT.HOLDMSG)
PAGE.WAIT.FONT))
(HOLDINGWIDTH (STRINGWIDTH (fetch (PAGEHOLDMSG MSG) of PAGE.WAIT.RELEASEMSG)
PAGE.WAIT.FONT))
(HoldMessageAdjustment 4)
WINDOW HoldingButtonWidth)
(OR ACTIVITY (SETQ ACTIVITY PAGE.WAIT.ACTIVITY))
(SETQ HoldingButtonWidth (WIDTHIFWINDOW (IPLUS HoldMessageAdjustment HOLDINGWIDTH
HoldMessageAdjustment)
BORDERSIZE))
(SETQ WINDOW (CREATEW (create REGION
WIDTH _ HoldingButtonWidth
HEIGHT _ (HEIGHTIFWINDOW (TIMES (SELECTQ ACTIVITY
(WINKING 2)
1)
(FONTHEIGHT PAGE.WAIT.FONT))
NIL BORDERSIZE))
NIL BORDERSIZE T))
(DSPFONT PAGE.WAIT.FONT WINDOW)
(DSPTEXTURE GRAYSHADE WINDOW)
(WINDOWPROP WINDOW (QUOTE SHRINKFN)
(QUOTE DON'T))
(WINDOWPROP WINDOW (QUOTE RESHAPEFN)
(QUOTE DON'T))
(WINDOWPROP WINDOW (QUOTE BUTTONEVENTFN)
(QUOTE \PageHold.buttonEventFn))
(WINDOWPROP WINDOW (QUOTE END.OF.PAGE.HOLD)
T)
(RETURN (create PAGEHOLDBUTTON
TIMERS _ (to 3 collect (SETUPTIMER 0))
ACTIVITY _ ACTIVITY
BUTTONIMAGE _ WINDOW
PAGEHOLDBUTTONWIDTH _ (WINDOWPROP WINDOW (QUOTE WIDTH]))

```

## (\PageHold.GET

```

[LAMBDA (ACTIVITY)
  (OR [for L on (PROG1 (CDR \PAGEHOLDBUTTONS)
    )
    when (EQ ACTIVITY (fetch (PAGEHOLDBUTTON ACTIVITY) of (CAR L)))
    do (RETURN (PROG1 (CAR L)
      (RPLACA L NIL]
      (NEWRESOURCE PAGEHOLDBUTTON ACTIVITY])
  )
(DECLARE: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \PAGEHOLDBUTTONS)
)
(DECLARE: DONTVAL@LOAD DOCOPY
(MOVD? (QUOTE PAGEFULLFN)
(QUOTE OLDPAGEFULLFN)
NIL T)
(/MOVD (QUOTE END.OF.PAGE.HOLD)
(QUOTE PAGEFULLFN))
)
(DECLARE: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
(PUTPROPS PAGEHOLD COPYRIGHT ("Xerox Corporation" 1982 1983 1984 1985 1986))

```

(\* JonL "12-Nov-84 20:28")  
(\* Comment PPLossage)

---

FUNCTION INDEX

END.OF.PAGE.HOLD .....	2	\EOP.DO.BUTTON .....	2	\PageHold.doMenu .....	5	\PageHold.printMessage ..	4
MakePageHoldButton .....	5	\PageHold.buttonEventFn ..	4	\PageHold.GET .....	6		

---

VARIABLE INDEX

HoldButtonBottomLine .....	2	PAGE.WAIT.HOLDMSG .....	1	PAGE.WAIT.STOPMSG .....	1
HoldingButtonMenu .....	2	PAGE.WAIT.IGNORETYPEAHEAD .....	1	TitleBarHeight .....	2
PAGE.WAIT.ACTIVITY .....	1	PAGE.WAIT.RELEASEMSG .....	1		
PAGE.WAIT.FONT .....	1	PAGE.WAIT.SECONDS .....	1		

---

RECORD INDEX

PAGEHOLDBUTTON .....	2	PAGEHOLDMSG .....	2
----------------------	---	-------------------	---

---

MACRO INDEX

CTRLREALLYDOWN? .....	1	TYPEAHEADP .....	1
-----------------------	---	------------------	---

---