File created: 25-Mar-2024 12:53:34 **{DSK}<home>frank>il>notecards>system>NCUTILITIES.;9**

changes to: (FNS NC.UpperLeftCH# NC.ScrollToCH#)

previous date: 8-Mar-2024 15:43:51 {DSK}<home>frank>il>notecards>system>NCUTILITIES.;8

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ NCUTILITIESCOMS
       [

;;; UTILITIES

        (GLOBALVARS WHOLEDISPLAY NC.OffScreenPosition \NC.NoBreakInReportError)
        [INITVARS (\NC.NoBreakInReportError)
               (NC.OffScreenPosition '(1500 . 1500]


;;; Coercing functions

        (FNS NC.CoerceToCard NC.CardFromWindow)
        [DECLARE%: DONTEVAL@LOAD DOCOPY (P (NC.StoreAutoloadFnFile (FUNCTION NC.ConvertNoteFileVersion2To3)
                                                 'NCCONVERTVERSION2TO3
                                                 'NOTECARDSDIRECTORIES]


;;; Shorthand window fns

        (FNS WW FLASHW)


;;; Other stuff

        (GLOBALVARS NC.PrintMsgFont)
        (INITVARS (NC.PrintMsgFont (FONTCREATE 'HELVETICA 10)))
        (FNS GETWREGION NC.AskUserWithMenu NC.AskYesOrNo NC.YesP FILDIR-EARLIEST FILDIR-VERSION GETMOUSEX
             GETMOUSEY LOWERLEFT MBUTTON.NEXT.FIELD.AS.TEXT.OR.IMAGEOBJ NC.AskUser NC.ZapAskUserProcess
             NC.AskUserResetWindow NC.ClearMsg NC.ClearMsgNow NC.ClearMsgWaiter NC.ZapPromptWindowWaiter
             NC.GreyCard NC.IDFromNumber NC.MoveWindowOntoScreen NC.NotDaughterP NC.MarkerMatchesCardP
             NC.PlaceMarkerP NC.ReportError NC.PrintMsg DFIRSTREMOVE NC.HoldTTYProcess NC.GetShrunkenWin
             NC.CoerceToNoteFileStream NC.DrawInnerBox NC.UnionListsOfLinks NC.ParseString
             NC.AppendStringToStream NC.CoerceToInterestedWindow NC.UpperLeftCH# NC.ScrollToCH#)
        (FNS NC.UpdateConfig NC.UpdateSingleFile NC.SetStandardConfigProps MAKECONFIGPROPSCOMS)
        (PROP (FILETYPE MAKEFILE-ENVIRONMENT)
              NCUTILITIES)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA MAKECONFIGPROPSCOMS)
                                                                             (NLAML)
                                                                             (LAMA NC.PrintMsg])


;;; UTILITIES

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS WHOLEDISPLAY NC.OffScreenPosition \NC.NoBreakInReportError)
)

(RPAQ? \NC.NoBreakInReportError )

(RPAQ? NC.OffScreenPosition '(1500 . 1500))


;;; Coercing functions

(DEFINEQ

(NC.CoerceToCard
  [LAMBDA (CardIdentifier)                                             ; Edited  4-Jan-89 17:16 by rtk

;;; Get the Card object corresponding to CardIdentifier.

;;; BK 1-4-89  Add code to recognize display streams

    (LET (Window)
         (AND CardIdentifier (COND
                                ((NC.CardP CardIdentifier)
                                 CardIdentifier)
                                ((WINDOWP CardIdentifier)
                                 (NC.CardFromWindow CardIdentifier))
                                [(TEXTSTREAMP CardIdentifier)
                                 (COND
                                    ((STREAMPROP CardIdentifier 'NoteCardObject))
                                    ((WINDOWP (SETQ Window (NC.TEditWindow CardIdentifier)))
                                     (NC.CardFromWindow Window]
                                ([AND (STREAMP CardIdentifier)
                                      (WINDOWP (SETQ Window (WFROMDS CardIdentifier]
                                 (NC.CardFromWindow Window))
                                (T (NC.ReportError "NC.CoerceToCard " (CONCAT "Arg not Window or TextStream or
                                                                           Card:   " CardIdentifier])


(NC.CardFromWindow
  [LAMBDA (Window)                                              (* fgh%: "14-Nov-85 00:07")

          (* * fgh |11/13/85| Updated to handle Card object.)

    (WINDOWPROP Window 'NoteCardObject])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(NC.StoreAutoloadFnFile (FUNCTION NC.ConvertNoteFileVersion2To3)
       'NCCONVERTVERSION2TO3
       'NOTECARDSDIRECTORIES)
)


;;; Shorthand window fns

(DEFINEQ

(WW
  [LAMBDA (X Y)                                                 (* fgh%: " 2-Apr-84 15:15")
    (WHICHW X Y])

(FLASHW
  [LAMBDA (WIN? N FLASHINTERVAL SHADE)                          (* kirk%: "15-Jul-86 08:51")

          (* * kirk 15Jul86 Old name kept for historical and shorthand reasons)

    (FLASHWINDOW WIN? N FLASHINTERVAL SHADE])

)


;;; Other stuff

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.PrintMsgFont)
)

(RPAQ? NC.PrintMsgFont (FONTCREATE 'HELVETICA 10))

(DEFINEQ

(GETWREGION
  [LAMBDA (W NEWREGIONFN NEWREGIONFNDATA MINWIDTH MINHEIGHT)    ; Edited 12-Jun-90 13:25 by mitani
                                                                (* gets a region from a window)
    (PROG ((REG (GETREGION MINWIDTH MINHEIGHT NIL NEWREGIONFN NEWREGIONFNDATA)))
          (RETURN (CREATEREGION (IDIFFERENCE (fetch (REGION LEFT) of REG)
                                       (DSPXOFFSET NIL W))
                         (IDIFFERENCE (fetch (REGION BOTTOM) of REG)
                                (DSPYOFFSET NIL W))
                         (fetch (REGION WIDTH) of REG)
                         (fetch (REGION HEIGHT) of REG])


(NC.AskUserWithMenu
  [LAMBDA (MenuItems Message InterestedWindow DontCloseAtEndFlg FlashFlg)
                                                                ; Edited  8-Dec-88 11:09 by krivacic

;;; Puts up a prompt window containing Message with an attached menu containing MenuItems and returns the selected menu item.

    (LET ((Event (CREATE.EVENT 'AskUserWithMenu))
          Menu MenuWindow PromptWin)
         (SETQ PromptWin (NC.PrintMsg InterestedWindow T Message))
         (if FlashFlg
             then (FLASHWINDOW PromptWin))
         (WINDOWPROP PromptWin 'AskUserWithMenu-Event Event)
         (SETQ MenuWindow (ATTACHMENU (SETQ Menu (create MENU
                                                        ITEMS _ MenuItems
                                                        WHENSELECTEDFN _ [FUNCTION (LAMBDA (Item Menu Button)
                                                                                     (LET
                                                                                      [(Win (MAINWINDOW
                                                                                             (WFROMMENU Menu]
                                                                                      (WINDOWPROP Win
                                                                                             '
                                                                                     AskUserWithMenu-SelectedItem
                                                                                             Item)
                                                                                      (NOTIFY.EVENT
                                                                                       (WINDOWPROP Win

                                                                                              '
                                                                                            AskUserWithMenu-Event
                                                                                              ]
                                                        MENUFONT _ (FONTCREATE 'HELVETICA 14 'BOLD)
                                                        MENUBORDERSIZE _ 1
                                                        ITEMHEIGHT _ 20))
                                 PromptWin
                                 (COND
                                    ((EQ PromptWin PROMPTWINDOW)
                                     'BOTTOM)
                                    (T 'TOP))
                                 'LEFT))
         (ALLOW.BUTTON.EVENTS)
         (AWAIT.EVENT Event)
         (DELETEMENU Menu T)
         (DETACHWINDOW MenuWindow)
         (OR DontCloseAtEndFlg (NC.ClearMsg InterestedWindow T))

         ;; Return the selected menu item and trash that WINDOWPROP at the same time just to be safe.

         (WINDOWPROP PromptWin 'AskUserWithMenu-SelectedItem NIL])
```

```
(NC.AskYesOrNo
  [LAMBDA (Msg Prompt FirstTry ClearFirstFlg MainWindow DontCloseAtEndFlg DontClearAtEndFlg)
                                                                  ; Edited  8-Dec-88 18:35 by krivacic

;;; Get a Yes or No response from the user via keyboard or mouse.  Return T if Yes answer and NIL otherwise.

;;; kirk 3Feb86 tried unsuccessfully to put a RESETLST RESETSAVE around menu so aborts would close it.

;;; fgh 6/8/86 Now uses NC.AttachPromptWindow in place of GETPROMPTWINDOW

    (LET (Menu MenuWindow PromptWin)

;;; Determine and open the prompt window if necessary

         (SETQ PromptWin (NC.PrintMsg MainWindow ClearFirstFlg Msg))

;;; Attach a yes/no menu to the prompt window

         (SETQ MenuWindow (ATTACHMENU [SETQ Menu (create MENU
                                                        ITEMS _ '(Yes No)
                                                        WHENSELECTEDFN _ [FUNCTION (LAMBDA (Item Menu Button)
                                                                                     (TTY.PROCESS
                                                                                      (WINDOWPROP
                                                                                       (MAINWINDOW (WFROMMENU
                                                                                                    Menu))
                                                                                      'PROCESS))
                                                                                     (BKSYSBUF
                                                                                      (CONCAT Item (CHARACTER
                                                                                                    13]
                                                        MENUFONT _ (FONTCREATE 'HELVETICA 14 'BOLD)
                                                        MENUBORDERSIZE _ 1
                                                        ITEMHEIGHT _ 20
                                                        ITEMWIDTH _ (TIMES 2 (STRINGWIDTH "Yes"
                                                                                    (FONTCREATE 'HELVETICA 14
                                                                                           'BOLD]
                                 PromptWin
                                 (COND
                                    ((EQ PromptWin PROMPTWINDOW)
                                     'BOTTOM)
                                    (T 'TOP))
                                 'LEFT))

;;; print msg in prompt window and wait for user repsonse

         (NC.YesP (PROG1 (NC.AskUser "" Prompt FirstTry NIL MainWindow T DontClearAtEndFlg T)
                      (DELETEMENU Menu T)
                      (DETACHWINDOW MenuWindow)
                      (if (OR DontCloseAtEndFlg (EQ PromptWin PROMPTWINDOW))
                        else (CLOSEW PromptWin))))])
```

```
(NC.YesP
  [LAMBDA (Answer)                                                (* fgh%: "19-Dec-85 17:03")

          (* * Is Answer a "yes" ?)

    (if (FMEMB (MKATOM Answer)
              '(Yes Y y YES yes))
        then T])
```

```
(FILDIR-EARLIEST
  [LAMBDA (FileSpec)
    (FILDIR-VERSION FileSpec 'EARLIEST])

(FILDIR-VERSION
  [LAMBDA (FileSpec Version)                                      (* edited%: "30-SEP-83 15:38")
    (PROG [FileName FileVersion Entry ResultsList (LatestFlag (COND
                                                                 ((EQ Version 'LATEST)
                                                                  T)
                                                                 (T NIL]
          [for File in (FILDIR FileSpec) do (SETQ FileName (UNPACKFILENAME File))
                                            (SETQ FileVersion (LISTGET FileName 'VERSION))
                                            (LISTPUT FileName 'VERSION NIL)
                                            (SETQ FileName (PACKFILENAME FileName))
                                            (SETQ Entry (FASSOC FileName ResultsList))
                                            (COND
                                               ((NULL Entry)
                                                (SETQ ResultsList (CONS (CONS FileName FileVersion)
                                                                        ResultsList)))
                                               ((AND LatestFlag (IGREATERP FileVersion (CDR Entry)))
                                                (RPLACD Entry FileVersion))
                                               ((ILESSP FileVersion (CDR Entry))
                                                (RPLACD Entry FileVersion]
          (RETURN (for File in (DREVERSE ResultsList) collect (PACK* (CAR File)
                                                                      ';
                                                                      (CDR File])

(GETMOUSEX
  [LAMBDA NIL                                                     (* fgh%: " 1-Apr-84 13:18")
    (GETMOUSESTATE)
    LASTMOUSEX])

(GETMOUSEY
  [LAMBDA NIL                                                     (* fgh%: " 1-Apr-84 13:18")
    (GETMOUSESTATE)
    LASTMOUSEY])

(LOWERLEFT
  [LAMBDA (Region)                                                (* fgh%: "30-Mar-84 20:01")
    (create POSITION
           XCOORD _ (fetch LEFT of Region)
           YCOORD _ (fetch BOTTOM of Region])

(MBUTTON.NEXT.FIELD.AS.TEXT.OR.IMAGEOBJ
  [LAMBDA (TEDIT.THING CH#)                                       (* fgh%: "31-May-84 18:06")

          (* Returns the first IMAGEOBJ in the next field. IF no such beast, returns the next field as text)

    (LET* ((TEXTOBJ (TEXTOBJ TEDIT.THING))
           (TEXTSTREAM (TEXTSTREAM TEXTOBJ)))
          (COND
             ((MBUTTON.FIND.NEXT.FIELD TEXTOBJ CH#)
              (\SETUPGETCH (fetch (SELECTION CH#) of (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
                     TEXTOBJ)
              (COND
                 ((CAR (bind ImageObj for CHNO from 1 to (fetch (SELECTION DCH) of (fetch (TEXTOBJ SCRATCHSEL)
                                                                                      of TEXTOBJ))
                          when (SETQ ImageObj (IMAGEOBJP (BIN TEXTSTREAM))) collect ImageObj)))
                 (T (replace (SELECTION SET) of (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ) with T)
                    (TEDIT.SEL.AS.STRING TEXTSTREAM (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ])

(NC.AskUser
  [LAMBDA (Msg Prompt FirstTry ClearFirstFlg MainWindow DontCloseAtEndFlg DontClearAtEndFlg PROMPTFORWORDFlg)
                                                                  ; Edited 14-Dec-88 14:06 by krivacic
                                                                  ; Get a response from the user --- using the promptwindow
                                                                  ; attached to MainWindow

;;; rht 9/16/84: Added DontClearAtEndFlg which if non-nil prevents the call to NC.ClearMsg.

;;; rht 5/22/85: Now uses TTYIN instead of PROMPTFORWORD so that people can edit their answer.

;;; rht 5/30/85: Added PROMPTFORWORDFlg so callers doing yes/no questions can get old style PROMPTFORWORD functionality.

;;; rht 8/6/85: Wrapped TTYIN with NLSETQ so wouldn't break if CR inadvertantly inserted in string.

;;; fgh 5/22/86 Added handling of COPYBYBKSYSBUF so that user can shift select into prompt windows being called from TEdit main windows.

;;; fgh 6/27/86 Added ERROR! is problems under TTYIN NLSETQ.  Allows other process to kill the askuser process.

;;; rg 5/1/87 fixed problem that sometimes caused Exec window's process to get smashed
;;; rg 8/11/87 reset interrupts before calling TTYIN

    (LET (AskWindow TextObj)
         (RESETLST

;;; If MainWindow is a TEdit window, make sur we can shift select into the prompt window.

               [RESETSAVE NIL `(NC.AskUserResetWindow ,(SETQ AskWindow (NC.PrintMsg MainWindow ClearFirstFlg Msg))
                                      ,MainWindow
                                      ,DontClearAtEndFlg
                                      ,DontCloseAtEndFlg]
               (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
               ;; KEEP old code to reset & save the copy of TEDIT formatting info   (* if (SETQ TextObj (WINDOWPROP MainWindow
                                                                                    (QUOTE TEXTOBJ))) then (RESETSAVE
                                                                                    (TEXTPROP TextObj (QUOTE COPYBYBKSYSBUF) T)
                                                                                    (BQUOTE (TEXTPROP (\, TextObj) COPYBYBKSYSBUF
                                                                                    (\, (TEXTPROP TextObj (QUOTE COPYBYBKSYSBUF)))))))

               ;; Disable Formatting on text only

               [if (SETQ TextObj (WINDOWPROP AskWindow 'TEXTOBJ))
                   then (RESETSAVE (TEXTPROP TextObj 'COPYBYBKSYSBUF T)
                                  `(TEXTPROP ,TextObj COPYBYBKSYSBUF ,(TEXTPROP TextObj 'COPYBYBKSYSBUF]

;;; Go ahead and ask

               (WINDOWPROP AskWindow 'AskUserProcess (THIS.PROCESS))
               (WINDOWADDPROP AskWindow 'CLOSEFN (FUNCTION NC.ZapAskUserProcess)
                      T)
               (PROG1 [if PROMPTFORWORDFlg
                          then (TTY.PROCESS (THIS.PROCESS))
                               [RESETFORM (RESET.INTERRUPTS (LISPINTERRUPTS)
                                                 T)
                                      (PROMPTFORWORD Prompt FirstTry "To type a ?, type CTRL-V followed by a ?."
                                             AskWindow NIL NIL (CHARCODE (EOL]
                        else (RESETLST
                                 (RESETSAVE (TTYDISPLAYSTREAM AskWindow))
                                 (RESETSAVE (RESET.INTERRUPTS (LISPINTERRUPTS)
                                                   T))
                                 (CAR (OR [NLSETQ (TTYIN (MKLIST (OR Prompt ""))
                                                         NIL NIL '(STRING NORAISE)
                                                         NIL NIL (AND FirstTry (LIST FirstTry]
                                          (ERROR!))))]
                   (WINDOWDELPROP AskWindow 'CLOSEFN (FUNCTION NC.ZapAskUserProcess))))])
```

(NC.ZapAskUserProcess

```
  [LAMBDA (Window)                                            (* Randy.Gobbel "12-Aug-87 16:04")
    (LET [(AskUserProcess (WINDOWPROP WINDOW 'AskUserProcess]
         (WINDOWDELPROP Window 'CLOSEFN (FUNCTION NC.ZapAskUserProcess))
         [AND (PROCESSP AskUserProcess)
              (PROCESS.EVAL AskUserProcess '(ERROR!]
         'DON'T])
```

(NC.AskUserResetWindow

```
  [LAMBDA (AskWindow MainWindow DontClearAtEndFlg DontCloseAtEndFlg)
                                                              ; Edited 13-Dec-88 14:57 by krivacic

;;; called from RESETRESTORE to blow away prompt window, unless flags tell us not to

;;; rg 3/6/87 created

;;; rg 5/1/87 now check AskWindow non-nil so we don't accidentally smash the Exec window

;;; rg 8/11/87 always blow away window on error

;;; Setting the PromptWindow PROCESS to NIL is to break a circularity caused by TEXTOBJ -> PROMPTWINDOW -> PROCESS -> TEXTSTREAM ->
;;; TEXTOBJ

    (AND AskWindow (WINDOWPROP AskWindow 'PROCESS NIL))
    (AND AskWindow (WINDOWDELPROP AskWindow 'CLOSEFN (FUNCTION NC.ZapAskUserProcess)))
    (if (EQ RESETSTATE 'ERROR)
        then (NC.ClearMsg MainWindow T)
      else (OR DontClearAtEndFlg (NC.ClearMsg MainWindow (NULL DontCloseAtEndFlg])
```

(NC.ClearMsg

```
  [LAMBDA (MainWindow ClosePromptWindowFlg WaitMsecs)         ; Edited  9-Jun-88 17:16 by Randy.Gobbel
                                                              ; Clear and optionally close the prompt window for MainWindow.
                                                              ; Wait (at least) specified time if supplied

;;; rg 6/8/88: Created.  Old code now in NC.ClearMsgNow.

    (DECLARE (GLOBALVARS NC.PromptWindowMonitorLock))
    (WITH.MONITOR NC.PromptWindowMonitorLock
          0
          (if (NOT (WINDOWP MainWindow))
              then (SETQ MainWindow (GETPROMPTWINDOW)))
          [if [AND (NULL WaitMsecs)
                   (NOT (WINDOWPROP MainWindow 'ClearPromptWindowProcess]
              then ;; no waiting involved

                   (NC.ClearMsgNow MainWindow ClosePromptWindowFlg)
            else ;; window already has a waiter of needs one.  Must at least update close/clear parameters

                 [WINDOWPROP MainWindow 'ClosePromptWindowFlg (OR ClosePromptWindowFlg (WINDOWPROP MainWindow

                                                                                                 'ClosePromptWindowFlg
                                                                                                 ]
                 [if (NUMBERP WaitMsecs)
                     then ;; set time for action, overriding any previously specified value

                          (WINDOWPROP MainWindow 'ClearMsgTimer (SETUPTIMER WaitMsecs (WINDOWPROP MainWindow
                                                                                                'ClearMsgTimer]
                 (if (NOT (WINDOWPROP MainWindow 'ClearPromptWindowProcess))
                     then (WINDOWPROP MainWindow 'ClearMsgEvent (CREATE.EVENT 'ClearMsgEvent))
                          (WINDOWADDPROP MainWindow 'CLOSEFN (FUNCTION NC.ZapPromptWindowWaiter))
                          (WINDOWPROP MainWindow 'ClearPromptWindowProcess (ADD.PROCESS `(NC.ClearMsgWaiter
                                                                                          ,MainWindow])])


(NC.ClearMsgNow
  [LAMBDA (MainWindow ClosePromptWindowFlg)                          ; Edited  9-Jun-88 14:25 by Randy.Gobbel
                                                                     ; Clear and optionally close the prompt window for MainWindow

;;; rht 9/10/85: Fixed so closes instead of removing prompt window and clears whether prompt win gets closed or not.

;;; rht 11/7/85: Now checks if MainWindow is a NC window.  If so, clears its promptwindow.  If not, clears MainWindow itself.

;;; fgh 5/1/86 Fix to above fix.  If clearing MainWindow, still need to check the ClosePromptWindowFlg.  Don't close the main window if this flag in NIL.

;;; fgh 6/8/86 Added call to REPOSITIONATTACHEDWINDOWS

;;; fgh 6/13/86 Now uses prompt window for NF menus as well as cards.

;;; fgh 6/27/86 Updated to match window width changes in NC.PrintMsg

;;; fgh&rht 7/4/86: Now checks that window isn't shrunken before calling REPOSITIONATTACHEDWINDOWS

;;; fgh 7/5/86 Redid the previous fix.  Replaced REPOSITIONATTACHEDWINDOWS with FREEATTACHWINDOW in order to handle cases for open but
;;; buried windows as well as shrunken windows.

;;; rht&rg&pmi: 4/22/87: Now smashes the PromptWindowProcess windowprop.

;;; rg 6/8/88: Renamed from NC.ClearMsg, which now takes a wait time arg.

    ;; WARNING! This procedure should only be called from inside NC.PromptWindowMonitorLock!

    (LET (PromptWindow)
         (COND
            [(WINDOWP MainWindow)
             (COND
                [(NC.NoteCardsWindowP MainWindow)
                 (SETQ PromptWindow (NC.GETPROMPTWINDOW MainWindow NIL NIL T))
                 (if (OPENWP PromptWindow)
                     then (CLEARW PromptWindow))
                 (WINDOWPROP PromptWindow 'MaxLineWidth NIL)
                 (WINDOWPROP PromptWindow 'LastLineLength NIL)
                 (WINDOWPROP PromptWindow 'OldMaxWidth 0)
                 (WINDOWPROP MainWindow 'PromptWindowLastNotCR NIL)
                 (COND
                    (ClosePromptWindowFlg (WINDOWPROP PromptWindow 'PromptWindowProcess NIL)
                           [WINDOWDELPROP MainWindow 'OtherPromptWindows (ASSOC PromptWindow (WINDOWPROP
                                                                                              MainWindow

                                                                                              '
                                                                                              OtherPromptWindows
                                                                                              ]
                           (FREEATTACHEDWINDOW PromptWindow)
                           (REMOVEWINDOW PromptWindow]
                (T (CLEARW MainWindow)
                   (if ClosePromptWindowFlg
                       then (CLOSEW MainWindow]
            (T (CLRPROMPT])


(NC.ClearMsgWaiter
  [LAMBDA (Window)                                                   ; Edited  9-Jun-88 17:23 by Randy.Gobbel
    (LET (WakeupTime ClosePromptWindowFlg)
         (WITH.MONITOR NC.PromptWindowMonitorLock
             (while [NOT (TIMEREXPIRED? (WINDOWPROP Window 'ClearMsgTimer]
                do (MONITOR.AWAIT.EVENT NC.PromptWindowMonitorLock (WINDOWPROP Window 'ClearMsgEvent)
                          (WINDOWPROP Window 'ClearMsgTimer)
                          T))
             (WINDOWDELPROP Window 'CLOSEFN (FUNCTION NC.ZapPromptWindowWaiter))
             (WINDOWPROP Window 'ClearMsgEvent NIL)
             (WINDOWPROP Window 'ClearMsgTimer NIL)
             (WINDOWPROP Window 'ClearPromptWindowProcess NIL)
             (SETQ ClosePromptWindowFlg (WINDOWPROP Window 'ClosePromptWindowFlg NIL))
             (NC.ClearMsgNow Window ClosePromptWindowFlg))])

(NC.ZapPromptWindowWaiter
  [LAMBDA (Window)                                              ; Edited  9-Jun-88 17:16 by Randy.Gobbel
    (WITH.MONITOR NC.PromptWindowMonitorLock
        (LET [(PromptWindowWaiter (WINDOWPROP Window 'ClearPromptWindowProcess]
             (WINDOWDELPROP Window 'CLOSEFN (FUNCTION NC.ZapPromptWindowWaiter))
             (AND (PROCESSP PromptWindowWaiter)
                  (DEL.PROCESS PromptWindowWaiter))))])

(NC.GreyCard
  [LAMBDA (Card)                                                ; Edited 12-Sep-88 13:51 by pmi
                                                                ; Grey over the interior of a card to mark it as obsolete.

    ;; pmi 9/12/88: Now uses NCP.GrayShade in BITBLT instead of GRAYSHADE.

    (DECLARE (GLOBALVARS NCP.GrayShade))
    (PROG ((Window (NC.FetchWindow Card)))
          (AND (WINDOWP Window)
               (BITBLT NIL NIL NIL Window NIL NIL NIL NIL 'TEXTURE 'PAINT NCP.GrayShade))
          (RETURN T])

(NC.IDFromNumber
  [LAMBDA (Number)                                              (* fgh%: " 9-Apr-84 19:24")
    (PACK* (SUBATOM 'NC00000 1 (IDIFFERENCE 7 (NCHARS Number)))
           Number])

(NC.MoveWindowOntoScreen
  [LAMBDA (Window)                                              (* rht%: "21-Mar-87 18:08")
                                                                (* Make sure a window and all its attachments are on the
                                                                screen.)

          (* * rht 8/28/85%: Hacked to handle very big card windows that together with attached windows might be too big to fit on
          screen.)

          (* * kirk 12Feb86 Added room for scroll bars.)

          (* * rht 3/4/86%: Changed to use RELMOVEW because otherwise it screws up when there's an attached window on the left
          or bottom of the window.)

          (* * rht 3/19/87%: Undid Kirk's change of 12Feb86. No longer leaves room for scroll bars.)

    (PROG (OldWindowRegion NewWindowRegion OldTop OldBottom NewLeft NewTop)
          (if [AND (WINDOWP Window)
                   (NOT (SUBREGIONP WHOLEDISPLAY (SETQ OldWindowRegion (WINDOWREGION Window]
              then (SETQ NewWindowRegion (COPY OldWindowRegion))
                   (SETQ OldTop (fetch (REGION TOP) of NewWindowRegion))
                   [if (GREATERP (fetch (REGION RIGHT) of NewWindowRegion)
                                (fetch (REGION RIGHT) of WHOLEDISPLAY))
                       then (SETQ NewLeft (replace (REGION LEFT) of NewWindowRegion
                                                 with (DIFFERENCE (fetch (REGION RIGHT) of WHOLEDISPLAY)
                                                             (fetch (REGION WIDTH) of NewWindowRegion]
                   (if (MINUSP (OR NewLeft (fetch (REGION LEFT) of NewWindowRegion)))
                       then (replace (REGION LEFT) of NewWindowRegion with 0))
                   [if (MINUSP (SETQ OldBottom (fetch (REGION BOTTOM) of NewWindowRegion)))
                       then (replace (REGION BOTTOM) of NewWindowRegion with 0)
                            (SETQ NewTop (PLUS OldTop (MINUS OldBottom]
                   [if (GREATERP (OR NewTop (fetch (REGION TOP) of NewWindowRegion))
                                (fetch (REGION TOP) of WHOLEDISPLAY))
                       then (replace (REGION BOTTOM) of NewWindowRegion with (DIFFERENCE (fetch (REGION TOP)
                                                                                                     of WHOLEDISPLAY)
                                                                                              (fetch (REGION HEIGHT)
                                                                                                 of NewWindowRegion]
                   (RELMOVEW Window (create POSITION
                                           XCOORD _ (DIFFERENCE (fetch (REGION LEFT) of NewWindowRegion)
                                                           (fetch (REGION LEFT) of OldWindowRegion))
                                           YCOORD _ (DIFFERENCE (fetch (REGION BOTTOM) of NewWindowRegion)
                                                           (fetch (REGION BOTTOM) of OldWindowRegion])

(NC.NotDaughterP
  [LAMBDA (StartCard CandidateCard LinkPredicate CheckedCardList)
                                                                (* fgh%: "16-Nov-85 00:31")

          (* Returns T if CandidateCard is not on any path emenating from StartID.
          Only links for which LinkPredicate is true are checked. LinkPredicate defaults to all links.)

    (LET (ToLinks)
          (OR LinkPredicate (SETQ LinkPredicate (FUNCTION TRUE)))
          (SETQ ToLinks (NC.RetrieveToLinks StartCard))
          (SETQ CheckedCardList (CONS StartCard CheckedCardList))
          (for Link in ToLinks bind DestinationCard when (AND (PROGN (SETQ DestinationCard (fetch (Link
                                                                                                      DestinationCard
                                                                                                         )
                                                                                                of Link))
                                                                      (for Card in CheckedCardList
                                                                         never (NC.SameCardP Card DestinationCard)))
                                                              (APPLY* LinkPredicate Link))
             always (AND (NEQ CandidateCard DestinationCard)
                         (NC.NotDaughterP DestinationCard CandidateCard LinkPredicate CheckedCardList])
```

(NC.MarkerMatchesCardP

```
  [LAMBDA (MarkerImageObject Card)                                   (* rht%: "13-Oct-86 12:08")

           (* * Return non-nil if Marker is appropriate for Card.)

    (AND (NC.PlaceMarkerP MarkerImageObject)
         (STREQUAL (IMAGEOBJPROP MarkerImageObject 'OBJECTDATUM)
                (if (NC.FileBoxP Card T)
                    then "File Boxes"
                  else "Note Cards"])
```

(NC.PlaceMarkerP

```
  [LAMBDA (ImageObject)                                              (* fgh%: " 5-Mar-84 01:37")
    (AND ImageObject (EQ (IMAGEOBJPROP ImageObject 'PUTFN)
                         (FUNCTION NC.PlaceMarkerPutFn])
```

(NC.ReportError

```
  [LAMBDA (FromFunction Msg)                                         ; Edited 29-Dec-87 11:09 by Burwell

;;; fgh 9/4/86 If \NC.NoBreakInReportError is non-NIL just prints msg in prompt window and cause ERROR!.

;;; rht 3/2/87: coerced the function name to be an atom to fix a bug showing up in lyric.

;;; rht 5/8/87: No longer breaks if HELPFLAG is nil.

    ;; rht 12/29/87: changed BREAK1 to CL:BREAK as per Stan's suggestion

    (DECLARE (GLOBALVARS HELPFLAG PROMPTWINDOW \NC.NoBreakInReportError))
    (if HELPFLAG
        then (if \NC.NoBreakInReportError
                 then (FLASHW PROMPTWINDOW)
                      (CLRPROMPT)
                      (PROMPTPRINT "NoteCards Error")
                      (PROMPTPRINT Msg)
                      (ERROR!)
               else (CL:BREAK "~%%In ~S, ~A." FromFunction Msg])
```

(NC.PrintMsg

```
  [LAMBDA Msgs                                                       ; Edited  9-Dec-88 10:20 by krivacic

    ;; Print the msgs in the specified window.  First argument is a window to print msg in, second arg is flag telling whether to clear first, rest of
    ;; arguments are simply prin1'ed to the msg window

;;; rht 7/24/85: Now checks to be sure that window arg is a notecards window.  If not, doesn't use promptwindow, but prints directly to window.

;;; rht 9/11/85: Added call to NC.MoveWindowOntoScreen in case we're crammed near the top of screen.

;;; fgh 6/8/86 Now opens prompt window to width that matches longest line in Msg.

;;; fgh 6/13/86 Now attaches prompt window to NF menus as well as cards.

;;; fgh 6/26/86 Now tries to make the window an appropriate width to hold the whole msg.  Height remains a constant.

;;; rht 3/23/87: Now gets font from globalvar NC.PrintMsgFont.

    (DECLARE (GLOBALVARS NC.PrintMsgFont))
    (LET
     ((LineCount 1)
      (CR-Count 0)
      [LineHeight (+ (FONTPROP NC.PrintMsgFont 'ASCENT)
                     (FONTPROP NC.PrintMsgFont 'DESCENT]
      (NoteCardWinFlg NIL)
      PromptWin CachedStream Window MaxWidth LineMaxes)
     (SETQ Window (ARG Msgs 1))

;;; find the maximun line length in the given arg list.  Cached on the main window under the PromptWindowLastNotCR prop is the length of the previous
;;; line if it didn't end in a carriage return.  Need to add this to the length of the first line in this call.

     (SETQ MaxWidth
      (APPLY
       (FUNCTION MAX)
       (OR (bind [LastNotCR _ (if (AND Window (NULL (ARG Msgs 2)))
                                   then (WINDOWPROP Window 'PromptWindowLastNotCR]
              for Msg from 3 to Msgs
              join

;;; Parse each arg into a list of line lengths.  Set the LastNotCR to the length of the last line in the arg if the arg does not end in a CR.  This will be added
;;; to the length of the first line in the next arg whether it be in this call or the next call to Printmsg.

                   [if (STREQUAL (ARG Msgs Msg)
                             "")
                       then (LIST 0)
                     else (while (SETQ Char (GNC CopyString)) as Ctr from 1
                            bind Char (LastCR _ 0)
                                 (CopyString _ (SUBSTRING (ARG Msgs Msg)
                                                      1))
                                 (OriginalString _ (SUBSTRING (ARG Msgs Msg)
                                                           1))
                            when (EQ Char (CHARACTER 13))
                            collect (SETQ CR-Count (+ CR-Count 1))
                                   (LET [(LINE-WIDTH (PLUS (OR LastNotCR 0)
                                                           (STRINGWIDTH (OR (SUBSTRING OriginalString (ADD1 LastCR
                                                                                                              )
                                                                                    (SUB1 Ctr))
                                                                             "")
                                                                  NC.PrintMsgFont]
                                        (SETQ LastCR Ctr)
                                        (SETQ LastNotCR 0)
                                        (SETQ LineMaxes (CONS LINE-WIDTH LineMaxes))
                                        LINE-WIDTH)
                            finally ;; if the last line doesn't end in a CR, save its length as LastNotCR but be sure to put its length on the list
                                  ;; anyway in case this is the last call.

                                  (if (NOT (EQP Ctr LastCR))
                                      then (SETQ $$VAL (CONS (PLUS (OR LastNotCR 0)
                                                                   (STRINGWIDTH (OR (SUBSTRING OriginalString
                                                                                            (ADD1 LastCR)
                                                                                            (SUB1 Ctr))
                                                                                    "")
                                                                           NC.PrintMsgFont))
                                                              $$VAL))
                                           (SETQ LastNotCR (CAR $$VAL))
                                    else (SETQ LastNotCR]
              finally ;; Cache the last linelength if it didn't end in a CR, we'll need it to add to the first line next time.

                     (SETQ LineMaxes (CONS LastNotCR LineMaxes))
                     (if Window
                         then (WINDOWPROP Window 'PromptWindowLastNotCR LastNotCR)))
           0)))
     [if (IGREATERP Msgs 2)
         then

;;; Figure out the prompt window

              (if (NOT (WINDOWP Window))
                  then (SETQ PromptWin PROMPTWINDOW)
                elseif (NC.NoteCardsWindowP Window)
                  then (SETQ PromptWin (NC.AttachPromptWindow Window NIL NIL NIL T))
                       (SETQ NoteCardWinFlg 'NOTECARD-WINDOW)
                else (SETQ PromptWin Window)
                    (SETQ NoteCardWinFlg 'ATTACHED-PROMPT-WINDOW))

;;; If requested, reset all the width caches.

              (if (AND PromptWin (ARG Msgs 2))
                  then (WINDOWPROP PromptWin 'OldMaxWidth 300))

;;; Set window width to the max width of the current max width and the cached max width from previous lines since the last clear

              [WINDOWPROP PromptWin 'OldMaxWidth (SETQ MaxWidth (MAX MaxWidth (OR (WINDOWPROP PromptWin
                                                                                              'OldMaxWidth)
                                                                                      300]

;;; Actual width of window is this maxwidth, except that if window is open, don't make it smaller.

              (SELECTQ NoteCardWinFlg
                   (NOTECARD-WINDOW
                        (SETQ PromptWin (NC.AttachPromptWindow Window (if (OPENWP PromptWin)
                                                                           then (MAX MaxWidth (WINDOWPROP
                                                                                                    PromptWin
                                                                                                    'WIDTH))
                                                                         else MaxWidth)))
                        (NC.MoveWindowOntoScreen Window))
                   (ATTACHED-PROMPT-WINDOW
                        (LET ((NEW-HEIGHT 0)
                              (OVERFLOW-LINES (bind (WINDOW-WIDTH _ (WINDOWPROP PromptWin 'WIDTH)) for WIDTH
                                                   in LineMaxes count (GREATERP WIDTH WINDOW-WIDTH)))
                              [WINDOW-BORDER (ITIMES 2 (WINDOWPROP PromptWin 'BORDER]
                              OLD-REGION)
                            ;; Determine if the Prompt window must be Extended in Height to fit the message.
                            (AND (GREATERP MaxWidth (+ (WINDOWPROP PromptWin 'WIDTH)
                                                      WINDOW-BORDER))
                                 (SETQ LineCount (+ LineCount CR-Count OVERFLOW-LINES))
                                 (SETQ NEW-HEIGHT (+ (ITIMES LineCount LineHeight)
                                                    WINDOW-BORDER))
                                 (LESSP (WINDOWPROP PromptWin 'HEIGHT)
                                        NEW-HEIGHT)
                                 (SETQ OLD-REGION (WINDOWPROP PromptWin 'REGION))
                                 (SHAPEW PromptWin (CREATEREGION (fetch (REGION LEFT) of OLD-REGION)
                                                         (fetch (REGION BOTTOM) of OLD-REGION)
                                                         (fetch (REGION WIDTH) of OLD-REGION)
                                                         NEW-HEIGHT))
                                 (REDISPLAYW PromptWin))))
                  NIL)

;;; Print the msg.

              (RESETLST
                  (OR NoteCardWinFlg (RESETSAVE (TTYDISPLAYSTREAM PromptWin)))
                  [RESETSAVE NIL `(DSPFONT ,(DSPFONT NC.PrintMsgFont PromptWin)
                                        ,PromptWin]
                  (if (ARG Msgs 2)
                      then (CLEARW PromptWin))
                  (LINELENGTH 1000 PromptWin)
                  (if (NULL PromptWin)
                      then (HELP))
                  (for Msg from 3 to Msgs collect (PRIN1 (ARG Msgs Msg)
                                                         PromptWin)))]
     PromptWin])

(DFIRSTREMOVE
  [LAMBDA (X L)                                                  (* rht%: "27-Apr-85 19:13")

          (* * Like DREMOVE except only deletes the first occurrence of X in L.
          Also note that it doesn't rearrange the cons nodes making up the list like DREMOVE does in the case when X =
          (CAR L))

    (for RestOfList on L first (if (EQ X (CAR L))
                                   then (RETURN (CDR L)))
       do (COND
             ((EQ X (CADR RestOfList))
              (RPLACD RestOfList (CDDR RestOfList))
              (RETURN L])

(NC.HoldTTYProcess
  [LAMBDA NIL                                                    (* rht%: "22-May-85 14:58")

          (* * Grabs the TTY process until it is explicitly placed elsewhere.)

    (TTY.PROCESS (ADD.PROCESS '(PROGN (WAIT.FOR.TTY)
                                      (while (TTY.PROCESSP) do (BLOCK)))
                    'NAME
                    'TtyHolder])

(NC.GetShrunkenWin
  [LAMBDA (CardOrWindow)                                         (* fgh%: "16-Nov-85 00:15")

          (* * Return the shrunken icon window for IDOrWindow. Return NIL if no corresponding shrunken window on screen.)

          (* * rht 10/20/85%: Rewrote to use ICONWINDOW prop of unshrunken window rather than looking at all windows on the
          screen.)

    (LET [(Window (COND
                     ((WINDOWP CardOrWindow))
                     ((NC.CardP CardOrWindow)
                      (NC.FetchWindow CardOrWindow]
         (COND
            ((WINDOWP Window)
             (OPENWP (WINDOWPROP Window 'ICONWINDOW])

(NC.CoerceToNoteFileStream
  [LAMBDA (CardOrNoteFileOrStream)                               (* rht%: "23-Jan-86 17:40")

          (* * Get stream from whatever it is.)

    (COND
       ((STREAMP CardOrNoteFileOrStream))
       ((type? NoteFile CardOrNoteFileOrStream)
        (fetch (NoteFile Stream) of CardOrNoteFileOrStream))
       ((type? Card CardOrNoteFileOrStream)
        (fetch (NoteFile Stream) of (fetch (Card NoteFile) of CardOrNoteFileOrStream])
(NC.DrawInnerBox
  [LAMBDA (Left Bottom Width Height LineWidth Operation ImageStream Dashing SkipLeftEdgeFlg SkipRightEdgeFlg
                ScaledIconWidth ScaledIconHeight)                ; Edited 27-Jan-88 09:23 by pmi

    ;; Draw a box that fits exactly inside the region given.  Omit the left edge if SkipLeftEdgeFlg non-nil.

    ;; pmi & rht 2/10/87: Changed to not overwrite corners of the box.

    ;; pmi 2/11/87: Updated for Multi-line link icons.  If multi-line, does not draw box edge in upper left corner where bitmap is placed.

    ;; rht 10/28/87: Added Dashing argument and passed to calls to DRAWLINE.

    ;; pmi 11/3/87: Added SkipRightEdgeFlg for new cross-file link icons (with additional attached bitmap on right side of link icons.)

    ;; pmi 1/27/88: Did some minor tweaking to the lines of the box so that they won't overlap at all.

    (if (AND (GREATERP Width 0)
             (GREATERP Height 0))
        then (LET (TrueWidth TrueHeight HalfBorderWidth Offset Right Top)
                  (SETQ TrueWidth (SUB1 Width))
                  (SETQ TrueHeight (SUB1 Height))
                  (SETQ HalfBorderWidth (FIX (TIMES LineWidth 0.5)))
                  (SETQ Offset (if (EVENP LineWidth)
                                   then -1
                                 else 0))
                  (SETQ Right (PLUS Left TrueWidth))
                  (SETQ Top (PLUS Bottom TrueHeight))
                  (LET (CenterLeft CenterRight CenterBottom CenterTop InnerBottom InnerRight InnerTop InnerLeft)
                       (SETQ CenterLeft (PLUS Left HalfBorderWidth Offset))
                       (SETQ CenterRight (DIFFERENCE Right HalfBorderWidth))
                       (SETQ CenterBottom (PLUS Bottom HalfBorderWidth))
                       (SETQ CenterTop (DIFFERENCE Top HalfBorderWidth))
                       (SETQ InnerBottom (PLUS Bottom LineWidth))
                       (SETQ InnerRight (DIFFERENCE Right LineWidth))
                       (SETQ InnerTop (DIFFERENCE Top LineWidth))
                       (SETQ InnerLeft (PLUS Left LineWidth))
                       (if (EVENP LineWidth)
                           then (SETQ CenterBottom (SUB1 CenterBottom)))

                       ;; Draw the bottom line

                       (DRAWLINE Left CenterBottom Right CenterBottom LineWidth Operation ImageStream NIL
                              Dashing)

                       ;; Draw the right line

                       (if SkipRightEdgeFlg
                           then (if ScaledIconWidth
                                    then ;; Leave a break for the attached bitmap

                                         (DRAWLINE CenterRight InnerBottom CenterRight (DIFFERENCE Top
                                                                                              ScaledIconHeight)
                                                LineWidth Operation ImageStream NIL Dashing)
                                         (SETQ InnerRight (DIFFERENCE Right ScaledIconWidth))
                                  else ;;  Skip the right edge altogether

                                       (SETQ InnerRight Right))
                         else (DRAWLINE CenterRight InnerBottom CenterRight Top LineWidth Operation ImageStream
                                     NIL Dashing))

                       ;; Draw the left line

                       (if SkipLeftEdgeFlg
                           then (if ScaledIconWidth
                                    then ;; Leave a break for the attached bitmap

                                         (DRAWLINE CenterLeft (DIFFERENCE Top ScaledIconHeight)
                                                CenterLeft InnerBottom LineWidth Operation ImageStream NIL
                                                Dashing)
                                         (SETQ InnerLeft (PLUS Left ScaledIconWidth))
                                  else ;;  Skip the left edge altogether

                                       (SETQ InnerLeft Left))
                         else (DRAWLINE CenterLeft Top CenterLeft InnerBottom LineWidth Operation ImageStream NIL
                                     Dashing))

                       ;; Draw the top line

                       (DRAWLINE InnerLeft CenterTop InnerRight CenterTop LineWidth Operation ImageStream NIL
                              Dashing])
```

```
(NC.UnionListsOfLinks
  [LAMBDA (LinksList1 LinksList2)                                ; Edited  9-Dec-87 14:22 by Gobbel

    ;; Return a list containing links appearing in either LinksList1 and LinksList2.

    (if (GREATERP (LENGTH LinksList2)
               (LENGTH LinksList1))
        then                                                     ; Swap in order that first list be the longest.
             (CL:PSETQ LinksList1 LinksList2 LinksList2 LinksList1))
    (APPEND LinksList2 (for Link1 in LinksList1 unless (for Link2 in LinksList2 thereis (NC.SameLinkP Link1 Link2))
                         collect Link1])
```

```
(NC.ParseString
  [LAMBDA (String)                                                 (* pmi%: " 6-Feb-87 10:47")

            (* * Parses a sting into words and "white space." Example%: String = "This   is a  test   " would result in
            ("This" "   " "is" " " "test" "   "))

    (COND
       [(STRINGP String)
        (PROG (StringLength SpacesPtr CharsPtr SubString StringList)
              (SETQ StringLength (NCHARS String))
              (SETQ SpacesPtr 1)
              (SETQ CharsPtr 1)
              (while (ILEQ SpacesPtr StringLength) do

            (* * Gather up adjacent spaces)

                                                            (while (AND (EQ (NTHCHAR String SpacesPtr)
                                                                            '% )
                                                                        (ILEQ SpacesPtr StringLength))
                                                               do (SETQ SpacesPtr (ADD1 SpacesPtr)))
                                                            (if (SETQ SubString (SUBSTRING String CharsPtr (SUB1
                                                                                                             SpacesPtr
                                                                                                              )))
                                                                then (SETQ StringList (CONS SubString StringList)))
                                                            (SETQ CharsPtr SpacesPtr)

            (* * Gather up adjacent characters)

                                                            (while (AND (NEQ (NTHCHAR String CharsPtr)
                                                                             '% )
                                                                        (ILEQ CharsPtr StringLength))
                                                               do (SETQ CharsPtr (ADD1 CharsPtr)))
                                                            (if (SETQ SubString (SUBSTRING String SpacesPtr (SUB1
                                                                                                              CharsPtr
                                                                                                               )))
                                                                then (SETQ StringList (CONS SubString StringList)))
                                                            (SETQ SpacesPtr CharsPtr))
              (RETURN (REVERSE StringList]
       (T String])

(NC.AppendStringToStream
  [LAMBDA (Stream String BoldFlg)                                  (* rht%: "26-Jun-85 12:17")

            (* * Add the String to the end of the tedit Stream.)

            (* * rht 11/16/84%: Now calls TEDIT.LOOKS in any case, bold or no.)

            (* * rht 6/26/85%: Took out call to TEDIT.LOOKS and just stuck boldifying into call to TEDIT.INSERT.)

    (TEDIT.INSERT Stream String (ADD1 (TEDIT.NCHARS Stream))
           [FONTCOPY (TEXTPROP Stream 'FONT)
                  'FACE
                  (COND
                     (BoldFlg 'BRR)
                     (T 'MRR]
           T])

(NC.CoerceToInterestedWindow
  [LAMBDA (WinOrCardOrNoteFile)                                    ; Edited 26-Feb-88 14:35 by pmi

    ;; Try to return a window that can be used for NC.PrintMsg, NC.AskUser, etc.  If arg is an open win, use that.  If displayed card, then use its
    ;; window.  If notefile, then use its notefile icon if open.  Last hope is to use the session icon if open.  Returning nil means that main prompt window
    ;; will have to be used.

    ;; pmi 2/26/88: Now the session icon may be used even if it is shrunken.  Also now checks that a card is both valid and displayed before returning
    ;; it.

    (DECLARE (GLOBALVARS NC.NoteCardsIconWindow))
    (LET (Card NoteFile)
         (COND
            ((OPENWP WinOrCardOrNoteFile)
             WinOrCardOrNoteFile)
            [(AND (type? NoteFile WinOrCardOrNoteFile)
                  (OPENWP (NCP.NoteFileIconWindow WinOrCardOrNoteFile]
            ((AND (OR (NC.CardP WinOrCardOrNoteFile)
                      (TEXTSTREAMP WinOrCardOrNoteFile))
                  (NCP.ValidCardP (SETQ Card (NC.CoerceToCard WinOrCardOrNoteFile)))
                  (NCP.CardDisplayedP Card))
             (NC.FetchWindow Card))
            [(AND (NCP.ValidCardP Card)
                  (SETQ NoteFile (fetch (Card NoteFile) of Card))
                  (OPENWP (NCP.NoteFileIconWindow NoteFile]
            ((OPENWP NC.NoteCardsIconWindow))
            [(OPENWP (WINDOWPROP NC.NoteCardsIconWindow 'ICONWINDOW]
            (T NIL])
```

```
(NC.UpperLeftCH#
  [LAMBDA (TEditWindow)                                             ; Edited  5-Aug-88 18:18 by Trigg

    ;; Returns the character number of the first currently displayed character in a tedit window.

    ;; rht 7/21/88: Fixed problem where bitmap in upper left corner of window caused bitmap editor to be run under call to TEDIT.SELECT.  Did that by
    ;; changing SELOPERATION arg from 'NORMAL to 'COPY.  Also changed the REGION arg from (fetch (TEXTOBJ MOUSEREGION) of TextObj)
    ;; to 'COPY.

    ;; rht 8/5/88: Another attempt at fixing above problem.  Now REGION arg is 'LINE.

    ;;

    ;; fgh 2024-03-25 Rewrote to use new public API TEDIT.XYTOCH

    ;;

    (TEDIT.XYTOCH 'LEFT 'TOP TEditWindow])

(NC.ScrollToCH#
  [LAMBDA (TEditWindow CH#)                                         ; Edited  9-Jul-91 21:44 by jds

    ;;

    ;; Scroll TEdit Window so that CH# is on the top line.

    ;;

    ;; 2024-03-04 FGH  Revamped to use only TEdit public APIs.  Previous versions used all kinds of TEdit internals and was almost totally ripped off
    ;; from TEDIT.NORMALIZECARET.

    ;;

    (LET* [(TEXTSTREAM (TEXTSTREAM TEditWindow))
           (SEL (AND TEXTSTREAM (TEDIT.SETSEL TEXTSTREAM CH# 0 'LEFT]
          (if SEL
              then (TEDIT.NORMALIZECARET TEXTSTREAM NIL T)
                   T
            else NIL])
)

(DEFINEQ

(NC.UpdateConfig
  [LAMBDA (Directory)                                               ; Edited 27-Feb-89 16:02 by krivacic

;;; bk 2/27/89: backed out changes.

    (DECLARE (GLOBALVARS NC.InstallFlg NC.Files NC.AutoloadStubFns NC.NoUpdateFlg NOTECARDTYPESTUBS
                    NOTECARDSDIRECTORIES))
    (if (NOT NC.NoUpdateFlg)
        then (SETQ Directory (OR Directory (DIRECTORYNAME T)))
             (for File in NC.Files do (for Prop in (GETPROP File 'ConfigProps) when (LISTP Prop)
                                         do (PUTPROP File (CAR Prop)
                                                     (APPLY* (CADR Prop)
                                                            File Directory)))
                                      (PRINTOUT T File ": " (GETPROP File 'SourceFile)
                                             " "
                                             (GETPROP File 'CompiledFile)
                                             " "
                                             (GETPROP File 'Directories)
                                             T))
             (SETQ NC.AutoloadStubFns (for TypeName in NOTECARDTYPESTUBS bind Fn
                                         eachtime (SETQ Fn (PACK* 'NCAddStub. TypeName 'Card))
                                         when (GETD Fn) collect Fn])

(NC.UpdateSingleFile
  [LAMBDA (File)                                                    (* Randy.Gobbel " 2-Sep-87 14:07")
    [for Prop in (GETPROP File 'ConfigProps) when (LISTP Prop) do (PUTPROP File (CAR Prop)
                                                                         (APPLY* (CADR Prop)
                                                                                File
                                                                                (DIRECTORYNAME T T]
    (PRINTOUT T File ": " (GETPROP File 'SourceFile)
           " "
           (GETPROP File 'CompiledFile)
           " "
           (GETPROP File 'FileDate)
           T])

(NC.SetStandardConfigProps
  [LAMBDA (File)                                                    ; Edited 15-Jan-88 18:08 by Randy.Gobbel

    ;; rg 7/8/87 add necessary config props for a standard NoteCards component file

    ;; rg 1/15/88 set CDR of FILEDATES to LOADFNS, so source access works right

    ;; rg 1/15/88 zapped all but SourceFile and CompiledFile

    (PUTPROP File 'ConfigProps '(([SourceFile (LAMBDA (File Directory)
                                                (FINDFILE (PACKFILENAME 'BODY File)
                                                       T
```

```
                                                              (OR (EVAL (GETPROP File 'Directories))
                                                                  (LIST Directory]
                                          CompiledFile
                                          (LAMBDA (File Directory)
                                            (FINDFILE (PACKFILENAME 'BODY File 'EXTENSION COMPILE.EXT)
                                                   T
                                                   (OR (EVAL (GETPROP File 'Directories))
                                                       (LIST Directory])
```

```
(MAKECONFIGPROPSCOMS
  [NLAMBDA Files                                                    (* Randy.Gobbel "13-Jul-87 14:18")
    (LET [(Props (for File in Files join (for Val in (GETPROP File 'ConfigProps)
                                                collect (if (LISTP Val)
                                                            then (CAR Val)
                                                          else Val]
         (SETQ Props (INTERSECTION Props Props))
         (for Prop in Props collect (APPEND (LIST 'IFPROP Prop)
                                            Files])
)

(PUTPROPS NCUTILITIES FILETYPE :FAKE-COMPILE-FILE)

(PUTPROPS NCUTILITIES MAKEFILE-ENVIRONMENT (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA MAKECONFIGPROPSCOMS)

(ADDTOVAR NLAML )

(ADDTOVAR LAMA NC.PrintMsg)
)
```

{MEDLEY}<notecards>system>NCUTILITIES.;1 28-Jun-2024 18:35:17

-- Listed on 30-Jun-2024 13:57:11 --

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX