File created: *5-Dec-87 20:03:36* **{QV}<NOTECARDS>1.3L>LIBRARY>NCSCREEN.;1**

changes to: *(VARS NCSCREENCOMS)*

previous date: *3-May-87 22:05:14 {QV}<NOTECARDS>1.3K>LIBRARY>NCSCREEN.;3*

Read Table: *INTERLISP*

Package: *INTERLISP*

Format: *MCCS*

```
;;
;; Copyright (c) 1984, 1985, 1987 by Xerox Corporation.  All rights reserved.

(RPAQQ NCSCREENCOMS
        ((DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT NCSCREEN))

;;; Functions for laying out sets of cards. Also some handy utility functions for windows.

        (GLOBALVARS SCREEN.CascadeXSpace SCREEN.CascadeYSpace)
        (VARS (SCREEN.CascadeXSpace 3)
              (SCREEN.CascadeYSpace 8))
                                                ; Functions for laying out cards on the screen.
        (FNS SCREEN.LayoutCardsInSquare SCREEN.LayoutCardsInCascade SCREEN.GetCascadePosition
             SCREEN.LayoutCardsInSurround SCREEN.SurroundCardWithFileBoxContents)
                                                ; These move cards and windows.
        (FNS SCREEN.MoveCardToPos SCREEN.MoveWinToPos SCREEN.WinShrinkAndPlace)
                                                ; These compute corner positions for windows.
        (FNS SCREEN.WinLLCorner SCREEN.WinLRCorner SCREEN.WinULCorner SCREEN.WinURCorner)))

(DECLARE%: DONTCOPY

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

;;; Functions for laying out sets of cards. Also some handy utility functions for windows.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS SCREEN.CascadeXSpace SCREEN.CascadeYSpace)
)

(RPAQQ SCREEN.CascadeXSpace 3)

(RPAQQ SCREEN.CascadeYSpace 8)

;; Functions for laying out cards on the screen.

(DEFINEQ

(SCREEN.LayoutCardsInSquare
  [LAMBDA (StartPos Cards)                                  (* rht%: " 3-May-87 21:55")

            (* * Layout the windows of Cards in a square with the upper left corner of the mess at StartPos.)

    (LET ((NumCards (LENGTH Cards))
          NumCols)
         [SETQ NumCols (ADD1 (FIX (SQRT (SUB1 NumCards]
         (for Card in Cards bind (Pos ← StartPos)
                                 (Col ← 1)
                                 NextRowPos Win
            do (SCREEN.MoveCardToPos Card Pos)
               (SETQ Win (NCP.CardWindow Card))
               (if (EQ Col 1)
                   then (SETQ NextRowPos (SCREEN.WinLLCorner Win)))
               (if (EQ Col NumCols)
                   then (SETQ Col 1)
                        (SETQ Pos NextRowPos)
                 else (SETQ Col (ADD1 Col))
                      (SETQ Pos (SCREEN.WinURCorner Win)))
            finally (GIVE.TTY.PROCESS Win])

(SCREEN.LayoutCardsInCascade
  [LAMBDA (StartPos Cards)                                  (* rht%: " 3-Jan-85 19:54")
```

```
        (* * Layout the windows of Cards in a cascaded, overlapping deck with the upper left corner of the mess at StartPos.)

    (for Card in Cards bind (Pos ← StartPos)
                            Win
       when (NCP.ValidCard Card) do (SCREEN.MoveCardToPos Card Pos)
                                    (SETQ Win (NCP.CardWindow Card))
                                    (SETQ Pos (SCREEN.GetCascadePosition Win))
                                    (GIVE.TTY.PROCESS Win])


(SCREEN.GetCascadePosition
  [LAMBDA (Window)                                          (* rht%: " 3-May-87 22:05")

        (* * Returns the position near the upper left corner of Window to place the next window in a cascade.)

    (DECLARE (GLOBALVARS WindowTitleDisplayStream))
    (LET [(Pos (SCREEN.WinULCorner Window))
          (Region (WINDOWPROP Window 'REGION]
         (create POSITION
                 XCOORD ← (IPLUS (fetch (POSITION XCOORD) of Pos)
                                 (WINDOWPROP Window 'BORDER)
                                 SCREEN.CascadeXSpace)
                 YCOORD ← (IDIFFERENCE (IPLUS (fetch (REGION HEIGHT) of Region)
                                              (fetch (REGION BOTTOM) of Region))
                                       (IPLUS (FONTPROP WindowTitleDisplayStream 'HEIGHT)
                                              SCREEN.CascadeYSpace])


(SCREEN.LayoutCardsInSurround
  [LAMBDA (CenterCard Cards)                                (* rht%: " 3-May-87 22:03")

        (* * Arrange the given Cards around the CenterCard CenterCard.
        Returns the number of cards displayed.)

    (PROG (CenterRegion CenterWindow (RemainingCards Cards))
          (SETQ CenterWindow (OR (NCP.CardDisplayedP CenterCard)
                                 (NCP.OpenCard CenterCard)))
          (if (NULL RemainingCards)
              then (GIVE.TTY.PROCESS CenterWindow)
                   (RETURN (LENGTH Cards)))
          (SETQ CenterRegion (WINDOWPROP CenterWindow 'REGION))
          [for bind Card Window (Pos ← (SCREEN.WinURCorner CenterWindow))
                    Region
                    (RemainingEdge ← (fetch (REGION HEIGHT) of CenterRegion))
             until (OR (MINUSP RemainingEdge)
                       (NULL RemainingCards))
             do (SETQ Card (CAR RemainingCards))
                (SETQ RemainingCards (CDR RemainingCards))
                (SCREEN.MoveCardToPos Card Pos 'UL)
                [SETQ Pos (SCREEN.WinLLCorner (SETQ Window (NCP.CardWindow Card]
                (SETQ RemainingEdge (DIFFERENCE RemainingEdge (fetch (REGION HEIGHT) of (WINDOWREGION
                                                                                         Window]
          (if (NULL RemainingCards)
              then (GIVE.TTY.PROCESS CenterWindow)
                   (RETURN (LENGTH Cards)))
          [for bind Card Window (Pos ← (SCREEN.WinULCorner CenterWindow))
                    Region
                    (RemainingEdge ← (fetch (REGION HEIGHT) of CenterRegion))
             until (OR (MINUSP RemainingEdge)
                       (NULL RemainingCards))
             do (SETQ Card (CAR RemainingCards))
                (SETQ RemainingCards (CDR RemainingCards))
                (SCREEN.MoveCardToPos Card Pos 'UR)
                [SETQ Pos (SCREEN.WinLRCorner (SETQ Window (NCP.CardWindow Card]
                (SETQ RemainingEdge (DIFFERENCE RemainingEdge (fetch (REGION HEIGHT) of (WINDOWREGION
                                                                                         Window]
          (if (NULL RemainingCards)
              then (GIVE.TTY.PROCESS CenterWindow)
```

```
                (RETURN (LENGTH Cards)))
        [for bind Card Window (Pos ← (SCREEN.WinLLCorner CenterWindow))
                Region
                (RemainingEdge ← (fetch (REGION WIDTH) of CenterRegion))
           until (OR (MINUSP RemainingEdge)
                     (NULL RemainingCards))
           do (SETQ Card (CAR RemainingCards))
              (SETQ RemainingCards (CDR RemainingCards))
              (SCREEN.MoveCardToPos Card Pos 'UL)
              [SETQ Pos (SCREEN.WinURCorner (SETQ Window (NCP.CardWindow Card]
              (SETQ RemainingEdge (DIFFERENCE RemainingEdge (fetch (REGION WIDTH) of (WINDOWREGION
                                                                                      Window]
        (GIVE.TTY.PROCESS CenterWindow)
        (RETURN (DIFFERENCE (LENGTH Cards)
                            (LENGTH RemainingCards])


(SCREEN.SurroundCardWithFileBoxContents
  [LAMBDA (Card Box)                                    (* rht%: "11-Feb-85 14:34")

        (* * Layout the children of Box around Card. If either arg is not present, ask user to select.)

    (PROG (Selection FileBoxChildren NumberOfChildren NumberDisplayed)
        [COND
            ((AND (NULL Card)
                  (NULL Box))
             (NCP.PrintMsg NIL T "Select center card and a filebox." (CHARACTER 13))
             (OR (SETQ Selection (NCP.SelectCards))
                 (RETURN))
             (SETQ Card (CAR Selection))
             (SETQ Box (CADR Selection)))
            ((NULL Card)
             (NCP.PrintMsg NIL T "Select center card.")
             (OR (SETQ Card (CAR (NCP.SelectCards)))
                 (RETURN)))
            ((NULL Box)
             (NCP.PrintMsg NIL T "Select FileBox.")
             (OR (SETQ Box (CAR (NCP.SelectCards)))
                 (RETURN]
        (SETQ FileBoxChildren (NCP.FileBoxChildren Box))
        (SETQ NumberOfChildren (LENGTH FileBoxChildren))
        (SETQ NumberDisplayed (SCREEN.LayoutCardsInSurround Card FileBoxChildren))
        (if (ILESSP NumberDisplayed NumberOfChildren)
            then (NC.PrintMsg NIL T "Only able to display " NumberDisplayed " out of " NumberOfChildren "
                              total cards." (CHARACTER 13])

)

;; These move cards and windows.

(DEFINEQ

(SCREEN.MoveCardToPos
  [LAMBDA (Card Pos Corner)                             (* rht%: " 2-May-87 18:37")

        (* * Move a card's window such that its Corner corner is at Pos.)

    (DECLARE (GLOBALVARS NC.OffScreenPosition))
    (SCREEN.MoveWinToPos (OR (NCP.CardWindow Card)
                             (NCP.OpenCard Card NC.OffScreenPosition))
          Pos Corner])


(SCREEN.MoveWinToPos
  [LAMBDA (Win Pos Corner)                              (* rht%: " 3-May-87 21:54")

        (* * Move a window such that its upper left corner is at Pos. Then possibly move it again to make sure it's all on the
        screen. Corner, if present, should be one of the litatoms UL, LL, UR, LR signifying which corner of the card should wind
        up at position Pos. Corner defaults to UL.)
```

```
    (LET ((Region (WINDOWREGION Win)))
         [SELECTQ Corner
                  (LL (MOVEW Win (fetch (POSITION XCOORD) of Pos)
                                 (fetch (POSITION YCOORD) of Pos)))
                  (UR [MOVEW Win (ADD1 (DIFFERENCE (fetch (POSITION XCOORD) of Pos)
                                                   (fetch (REGION WIDTH) of Region)))
                                 (ADD1 (DIFFERENCE (fetch (POSITION YCOORD) of Pos)
                                                   (fetch (REGION HEIGHT) of Region])
                  (LR (MOVEW Win (ADD1 (DIFFERENCE (fetch (POSITION XCOORD) of Pos)
                                                   (fetch (REGION WIDTH) of Region)))
                                 (fetch (POSITION YCOORD) of Pos)))
                  (MOVEW Win (fetch (POSITION XCOORD) of Pos)
                             (ADD1 (DIFFERENCE (fetch (POSITION YCOORD) of Pos)
                                               (fetch (REGION HEIGHT) of Region]
         (SETQ Region (MAKEWITHINREGION (COPY (WINDOWREGION Win))
                                        WHOLESCREEN))
         (MOVEW Win (fetch (REGION LEFT) of Region)
                    (fetch (REGION BOTTOM) of Region])


(SCREEN.WinShrinkAndPlace
  [LAMBDA (Win)                                          (* rht%: " 2-May-87 18:39")

          (* * Shrink the given window, if not already shrunk, and place in the upper left corner of Win.)

    (DECLARE (GLOBALVARS NC.OffScreenPosition))
    (LET ((WinULCorner (SCREEN.WinULCorner Win))
          ShrunkWin)
         (if (AND Win (SETQ ShrunkWin (SHRINKW Win NIL NC.OffScreenPosition)))
             then (SCREEN.MoveWinToPos ShrunkWin WinULCorner)
                  ShrunkWin
           else Win])

)

;; These compute corner positions for windows.

(DEFINEQ

(SCREEN.WinLLCorner
  [LAMBDA (Win)                                          (* rht%: " 2-May-87 18:38")

          (* * Return the position of the lower left corner of Win.)

    (LET ((Region (WINDOWREGION Win)))
         (create POSITION
                 XCOORD ← (fetch (REGION LEFT) of Region)
                 YCOORD ← (fetch (REGION BOTTOM) of Region])


(SCREEN.WinLRCorner
  [LAMBDA (Win)                                          (* rht%: " 2-May-87 18:42")

          (* * Return the position of the lower right corner of Win.)

    (LET ((Region (WINDOWREGION Win)))
         (create POSITION
                 XCOORD ← (fetch (REGION RIGHT) of Region)
                 YCOORD ← (fetch (REGION BOTTOM) of Region])


(SCREEN.WinULCorner
  [LAMBDA (Win)                                          (* rht%: " 2-May-87 18:42")

          (* * Return the position of the upper left corner of Win.)

    (LET ((Region (WINDOWREGION Win)))
```

```
        (create POSITION
                XCOORD ← (fetch (REGION LEFT) of Region)
                YCOORD ← (fetch (REGION TOP) of Region])


(SCREEN.WinURCorner
  [LAMBDA (Win)                                          (* rht%: " 2-May-87 18:43")

            (* * Return the position of the upper right corner of Win.)

    (LET ((Region (WINDOWREGION Win)))
        (create POSITION
                XCOORD ← (fetch (REGION RIGHT) of Region)
                YCOORD ← (fetch (REGION TOP) of Region])

)

(PUTPROPS NCSCREEN COPYRIGHT ("Xerox Corporation" 1984 1985 1987))
```

FUNCTION INDEX

SCREEN.GetCascadePosition 2
SCREEN.LayoutCardsInCascade 1
SCREEN.LayoutCardsInSquare 1
SCREEN.LayoutCardsInSurround 2
SCREEN.MoveCardToPos 3
SCREEN.MoveWinToPos 3
SCREEN.SurroundCardWithFileBoxContents ...3
SCREEN.WinLLCorner 4
SCREEN.WinLRCorner 4
SCREEN.WinShrinkAndPlace 4
SCREEN.WinULCorner 4
SCREEN.WinURCorner 5

VARIABLE INDEX

NCSCREENCOMS 1
SCREEN.CascadeXSpace 1
SCREEN.CascadeYSpace 1

PROPERTY INDEX

NCSCREEN 1