```
(RPAQQ NCBITMAPCARDCOMS
       ((* * This file implements a new NoteCards card type. This is the BITMAP card allowing the user to keep
          end edit bitmaps without much of the overhead associated with bitmaps in text or sketch cards. Bitmap
          card substances are lists in which the CAR is a bitmap and the CDR is an ASSOC-list of links and
          positions for those links within the card.)
        (* * Originally written by Dave Newman)
        (* * Heavily modified and extended by Len Gaska. A separate and much more extensive bitmap editor was
          implemented and interated into the code)
        (* * Basic Functions)
        (FNS NC.BitmapCardCloseFn NC.BitmapCardCollectLinks NC.BitmapCardCopySubstance NC.BitmapCardCreate
          NC.BitmapCardDeleteLinks NC.BitmapCardDirtyP NC.BitmapCardEdit NC.BitmapCardGetSubstance
          NC.BitmapCardInsertLinks NC.BitmapCardLinkFromPosition NC.BitmapCardMarkDirty
          NC.BitmapCardPutSubstance NC.BitmapCardSelectLink NC.BitmapCardUpdateLinks
          NC.BitmapCardTranslateWindowPosition)
        (* * Auxiliary Functions)
        (FNS NC.BitmapCardButtonEventFn NC.BitmapCardCopyButtonEventFn NC.BitmapCardDisplayLinkIcon
          NC.BitmapCardShrinkFn NC.BitmapCardShowAllLinks NC.BitmapCardWindowSetup)
        (* * Bitmap editing functions specific to the Bitmap card)
        (FNS NC.BitmapAddLink NC.BitmapDeleteLink NC.BitmapMoveLink)
        (* * Auxiliary editing functions.)
        (* * Temporary function until patch appears.)
        (FNS NC.AddLinkToCard)
        (* * Card Type Creation)
        (FNS NC.AddBitmapCardType)
        (* * Auxiliary files)
        (FILES NCTYPESMECH (SYSLOAD FROM LISPUSERS)
               IMAGEOBJ)
        (FILES BITMAPEDITOR)
        (* * Variables)
        (VARS NC.BitmapCardBitmap NC.BitmapCardIconBitmap NC.BitmapCardMenuItems)
        (GLOBALVARS NC.BitmapCardBitmap NC.BitmapCardIconBitmap NC.BitmapCardMenuItems)
        (* * Initialization)
        (P (NC.AddBitmapCardType)))))
```

(* * This file implements a new NoteCards card type. This is the BITMAP card allowing the user to keep end edit bitmaps without much of the overhead associated with bitmaps in text or sketch cards.
Bitmap card substances are lists in which the CAR is a bitmap and the CDR is an ASSOC-list of links and positions for those links within the card.)

(* * Originally written by Dave Newman)

(* * Heavily modified and extended by Len Gaska. A separate and much more extensive bitmap editor was implemented and interated into the code)

(* * Basic Functions)

```
(DEFINEQ
```

(NC.BitmapCardCloseFn
```
  [LAMBDA (CardOrWindowOrSubstance)                                    (* Gaska "26-Aug-88 14:35")

          (* * Close a bitmap card window.)

    (EditBitmapClose (NCP.CardWindow (NC.CoerceToCard CardOrWindowOrSubstance])
```

(NC.BitmapCardCollectLinks
```
  [LAMBDA (Card CheckAndDeleteFlg ReturnLinkIconsFlg ReturnLocationsFlg)
                                                                       (* Newman "21-Jul-88 13:20")

          (* * Return a list of links in a bitmap card.)

    (LET ((Window (NCP.CardWindow Card))
          (Substance (NC.FetchSubstance Card))
          DirtyFlg)
         (CONS (for ActualLink in (for Link&Position in (CDR Substance) collect (CAR Link&Position))
                  when (if (OR (NULL CheckAndDeleteFlg)
                               (AND (LISTP CheckAndDeleteFlg)
                                    (FMEMB (fetch (Link DestinationCard) of ActualLink)
                                           CheckAndDeleteFlg))
                               (NC.ValidLinkP ActualLink))
                         then                                          (* This link is OK, return T)
                          T
                         else (NCP.ReportError (QUOTE NC.BitmapCardCollectLinks)
```

```
                                     "Unable to fix bad links")
                                  (SETQ DirtyFlg T)                          (* This link is bad, set DirtyFlg and return NIL)
                                  NIL)
                       collect (COND
                              [(AND ReturnLinkIconsFlg (NOT ReturnLocationsFlg))
```

(* Can't return link icons without the window. If the window is not up, I am not sure what will happen.)

```
                                  (CDR (ASSOC ActualLink (WINDOWPROP Window (QUOTE LinkIcons]
                              [(AND ReturnLinkIconsFlg ReturnLocationsFlg)
                                                                   (* Return Link Icon and Position)
                                  (CONS [CDR (ASSOC ActualLink (WINDOWPROP Window (QUOTE LinkIcons]
                                        (CDR (ASSOC ActualLink (CDR Substance]
                              ((AND (NOT ReturnLinkIconsFlg)
                                    ReturnLocationsFlg)                      (* Return Link and position)
                               (ASSOC ActualLink (CDR Substance)))
                              (T                                            (* Return the link itself.)
                                 ActualLink)))
                       DirtyFlg])
```

(NC.BitmapCardCopySubstance
```
  [LAMBDA (Card FromStream ToStream Length)                          (* Newman "22-Apr-87 17:49")
```

(* * Copy the substance of a bitmap card from one stream to another.
Return T (This function is copied from the TEXT card implementation.))

```
    (LET* ((FromStartPtr (GETFILEPTR FromStream))
           (FromEndPtr (PLUS Length FromStartPtr)))
          (COPYBYTES FromStream ToStream FromStartPtr FromEndPtr))
    T])
```

(NC.BitmapCardCreate
```
  [LAMBDA (Card Title NoDisplayFlg Bitmap)                           (* Gaska "28-Aug-88 12:09")
```

(* * Create a blank bitmap card)

```
    (LET [(Region (if (NOT (BITMAPP Bitmap))
                      then (GETREGION)
                      else (CREATEREGION 0 0 (BITMAPWIDTH Bitmap)
                                         (BITMAPHEIGHT Bitmap]
```

(* * if we are not passed a bitmap, create and fill the bitmap for the card.)

```
         (if (NOT (BITMAPP Bitmap))
             then (SETQ Bitmap (BITMAPCREATE (fetch (REGION WIDTH) of Region)
                                             (fetch (REGION HEIGHT) of Region)))
                  (BITBLT ScreenBitMap (fetch (REGION LEFT) of Region)
                          (fetch (REGION BOTTOM) of Region)
                          Bitmap 0 0 (fetch (REGION WIDTH) of Region)
                          (fetch (REGION HEIGHT) of Region)))
```

(* * Set the substance of the card to a copy of the bitmap.)

```
         (NC.SetSubstance Card (LIST (EditBitmapCopyBitmap Bitmap)))
```

(* * Do or don't create and display the card window, and return the correct thing.)

```
         (if NoDisplayFlg
             then Card
            else (LET [(Window (NC.MakeNewCardWindow Card (OR Title "Untitled")
                               NIL NIL (GETBOXREGION (WIDTHIFWINDOW (fetch WIDTH of Region))
                                                     (HEIGHTIFWINDOW (fetch HEIGHT of Region)
                                                           T)
                                                     (IDIFFERENCE LASTMOUSEX (WIDTHIFWINDOW (fetch WIDTH
                                                                                                    of Region)))
                                                     LASTMOUSEY NIL "Position New Card Window."]
                       (EditBitmap Bitmap Window T NC.BitmapCardMenuItems)
                       (NC.BitmapCardWindowSetup Window (NC.RetrieveType Card)
                             (LIST Bitmap))
                       Window])
```

(NC.BitmapCardDeleteLinks
```
  [LAMBDA (SourceCard LinkOrDestinationCard Don'tCreateDeletedImageObjFlg)
                                                                   (* Gaska "29-Aug-88 07:29")
```

(* * Delete a link icon from a bitmap card.)

```
    (DECLARE (GLOBALVARS NC.NoDeleteImageFns NC.UseDeletedLinkIconIndicatorsFlg NC.DeletedLinkImageObject))
    (LET ((LinkFlg (type? Link LinkOrDestinationCard))
         DestinationCard Substance Window)
```

(* * Do we have a link or a Source and a Destination?)

```
         (COND
```

```
                    [LinkFlg (OR (NC.CardP SourceCard)
                                 (SETQ SourceCard (fetch (Link SourceCard) of LinkOrDestinationCard]
                    (T (SETQ DestinationCard LinkOrDestinationCard)))
```

(* * Get the Substance and Window)

```
          (SETQ Substance (NC.FetchSubstance SourceCard))
          (SETQ Window (NCP.CardWindow SourceCard))
```

(* * For each link in the card, delete it if so desired.)

```
          (for Link&Position in (CDR Substance) when [COND
                                                        (LinkFlg (NC.SameLinkP LinkOrDestinationCard (CAR
                                                                                                      Link&Position
                                                                                                      )))
                                                        (T (NC.SameCardP DestinationCard (fetch (Link
                                                                                                  DestinationCard
                                                                                                  )
                                                                                         of (CAR Link&Position]
             do
```

(* * Remove the Link from the substance)

```
                 (if (NULL (DREMOVE (ASSOC (CAR Link&Position)
                                          (CDR Substance))
                                   (CDR Substance)))
                     then (RPLACD Substance NIL))
```

(* * Remove the link from the LinkIcons ASSOC list on the window)

```
                 (if [NULL (DREMOVE (ASSOC (CAR Link&Position)
                                          (WINDOWPROP Window (QUOTE LinkIcons)))
                                   (WINDOWPROP Window (QUOTE LinkIcons]
                     then (WINDOWPROP Window (QUOTE LinkIcons)
                                      NIL))
                 (REDISPLAYW Window])
```

(**NC.BitmapCardDirtyP**
```
  [LAMBDA (Card)                                              (* Gaska "26-Aug-88 13:27")
```

(* * We store the property of dirtyness on the bitmap of the card.
So, this function returns the value of that property.)

```
     (WINDOWPROP (NCP.CardWindow Card)
           (QUOTE Dirty])
```

(**NC.BitmapCardEdit**
```
  [LAMBDA (Card Substance Position)                           (* Gaska "29-Aug-88 07:52")
```

(* * Bring up a bitmap card window)

```
     (if (NC.FetchWindow Card)
         then (LET ((Window (NC.FetchWindow Card)))
                   (TOTOPW Window)
                   (if (POSITIONP Position)
                       then (MOVEW Window Position)
                     elseif (REGIONP Position)
                       then (NCP.ReportError (QUOTE NC.BitmapCardEdit)
                                   "Attempt to reshape bitmap card window."))
                   (RPTQ 2 (FLASHW Window))
                   Window)
       else (LET ((Window (CREATEW (NC.DetermineDisplayRegion Card Position)
                                   (NC.RetrieveTitle Card)
                                   NIL T)))
                 (EditBitmap (CAR Substance)
                        Window T NC.BitmapCardMenuItems)
                 (NC.BitmapCardWindowSetup Window (NC.RetrieveType Card)
                        Substance)
                 [for Link&Icon in (WINDOWPROP Window (QUOTE LinkIcons))
                    do (NC.BitmapCardDisplayLinkIcon (CDR Link&Icon)
                              Window
                              (CDR (ASSOC (CAR Link&Icon)
                                          (CDR Substance]
                 Window])
```

(**NC.BitmapCardGetSubstance**
```
  [LAMBDA (Card Length Stream)                                (* Newman "22-Apr-87 18:55")
```

(* * Read a bitmap card's substance from the database stream.)

```
     (CONS (READBITMAP Stream)
           (LET ((Links (NC.ReadListOfLinks Stream))
                 (PositionList (READ Stream)))
                (for Link in Links as Position in PositionList collect (CONS Link Position])
```

⟮**NC.BitmapCardInsertLinks**
  [LAMBDA (Card Window LinkOrLinkLabel Position)   (* Gaska "18-Aug-88 09:17")

        (* * insert a link in a bitmap card.)

    (LET* [(Substance (NC.FetchSubstance (NC.CoerceToCard Window)))
           (LinkIcons (WINDOWPROP Window (QUOTE LinkIcons)))
           (Link (**if** (NCP.ValidLinkTypeP LinkOrLinkLabel (NCP.NoteFileFromCard Card))
                     then (NC.MakeLink Window LinkOrLinkLabel)
                  elseif (NCP.ValidLinkP LinkOrLinkLabel)
                     then LinkOrLinkLabel
                  else (ERROR LinkOrLinkLabel " not a link or a label.")))
           (LinkIcon (NC.MakeLinkIcon Link))
           (Position (OR (POSITIONP Position)
                     (GETBOXPOSITION (**fetch** (IMAGEBOX XSIZE) **of** (IMAGEBOXSIZE LinkIcon))
                          (**fetch** (IMAGEBOX YSIZE) **of** (IMAGEBOXSIZE LinkIcon))
                          NIL NIL Window]

        (* * Get the position of the Link Icon)

        (**if** (INSIDEP (DSPCLIPPINGREGION NIL Window)
                 Position)
           then

        (* * add the link to the card substance)

                (NCONC Substance (LIST (CONS Link Position)))

        (* * Make the Link Icon)

        (* * Put the link Icon on the Window's proplist)

                (COND
                   [(NULL LinkIcons)
                    (SETQ LinkIcons (LIST (CONS Link LinkIcon]
                   (T (PUTASSOC Link LinkIcon LinkIcons)))
                (WINDOWPROP Window (QUOTE LinkIcons)
                      LinkIcons)

        (* * DIsplay the link Icon)

                (NC.BitmapCardDisplayLinkIcon LinkIcon Window Position)
           else (NCP.DeleteLinks Link])

⟮**NC.BitmapCardLinkFromPosition**
  [LAMBDA (Window Position)   (* Gaska "28-Aug-88 12:15")

        (* * Return the link, if any, displayed at Position in Window. Dependent on the fact that
        (CAR NIL) is NIL.)

    (**bind** Link Region **for** Icon&Link **in** (WINDOWPROP Window (QUOTE LinkIcons))
       thereis [LET* ([LinkPosition (CDR (ASSOC (CAR Icon&Link)
                                              (CDR (NC.FetchSubstance (NC.CardFromWindow Window]
                    (ImageBox (NC.LinkIconImageBoxFn (CDR Icon&Link)
                                    Window)))
                    (SETQ Region (**create** REGION
                                       LEFT _ (CAR LinkPosition)
                                       BOTTOM _ (CDR LinkPosition)
                                       WIDTH _ (CAR ImageBox)
                                       HEIGHT _ (CADR ImageBox)))
                  (**if** (INSIDEP Region Position)
                      then (SETQ Link (CAR Icon&Link]
       finally (**if** Link
                then (RETURN (LIST Link Region])

⟮**NC.BitmapCardMarkDirty**
  [LAMBDA (Card ResetFlg)   (* Gaska "26-Aug-88 14:31")

        (* * Set the dirty flag on the bitmap card)

    (EditBitmapMakeDirty (NCP.CardWindow Card))

⟮**NC.BitmapCardPutSubstance**
  [LAMBDA (Card Stream)   (* Newman "22-Apr-87 19:13")

        (* * Store a bitmap card in the stream.)

    (LET ((Substance (NC.FetchSubstance Card)))

        (* * Write down the bitmap, links, and link positions for later retrieval)

        (PRINTBITMAP (CAR Substance)

```
                Stream)
            (NC.WriteListOfLinks Stream (for Link&Position in (CDR Substance) collect (CAR Link&Position)))
            (PRINT (for Link&Position in (CDR Substance) collect (CDR Link&Position))
                    Stream)
          0])
```

(NC.BitmapCardSelectLink
```
  [LAMBDA (Window)                                                        (* Gaska "26-Aug-88 13:14")

        (* * Return the link, if any, displayed at Position in Window.)

    (LET* ([LinkRegions (for Icon&Link in (WINDOWPROP Window (QUOTE LinkIcons))
                            collect (LET ([LinkPosition (CDR (ASSOC (CAR Icon&Link)
                                                                    (CDR (NC.FetchSubstance (NC.CardFromWindow
                                                                                                    Window]
                                         (ImageBox (NC.LinkIconImageBoxFn (CDR Icon&Link)
                                                                          Window)))
                                        (LIST (CAR Icon&Link)
                                              (create REGION
                                                      LEFT _ (fetch (POSITION XCOORD) of LinkPosition)
                                                      BOTTOM _ (fetch (POSITION YCOORD) of LinkPosition)
                                                      WIDTH _ (CAR ImageBox)
                                                      HEIGHT _ (CADR ImageBox]
           (DisplayStream (GETSTREAM Window)))
        (TOTOPW Window)
        (until [AND (MOUSESTATE (NOT UP))
                    (LET NIL (GETMOUSESTATE)
                         (EditBitmapInsideWindow Window (LASTMOUSEX DisplayStream)
                                 (LASTMOUSEY DisplayStream]
           do NIL)
        (bind InsideLink? SelectedLink until (LASTMOUSESTATE UP)
           do (GETMOUSESTATE)
              [LET* ((X (LASTMOUSEX DisplayStream))
                     (Y (LASTMOUSEY DisplayStream)))
                    (SETQ InsideLink? (for LinkInfo in LinkRegions thereis (INSIDEP (CADR LinkInfo)
                                                                                     X Y)))
                    (if InsideLink?
                        then (if (NULL (EQUAL SelectedLink InsideLink?))
                                 then (if SelectedLink
                                          then (DSPFILL (CADR SelectedLink)
                                                        BLACKSHADE
                                                        (QUOTE INVERT)
                                                        DisplayStream))
                                      (SETQ SelectedLink InsideLink?)
                                      (DSPFILL (CADR SelectedLink)
                                               BLACKSHADE
                                               (QUOTE INVERT)
                                               DisplayStream))
                        else (if SelectedLink
                                 then (DSPFILL (CADR SelectedLink)
                                               BLACKSHADE
                                               (QUOTE INVERT)
                                               DisplayStream)
                                      (SETQ SelectedLink NIL]
           finally (if SelectedLink
                       then (DSPFILL (CADR SelectedLink)
                                     BLACKSHADE
                                     (QUOTE INVERT)
                                     DisplayStream))
                   (RETURN SelectedLink])
```

(NC.BitmapCardUpdateLinks
```
  [LAMBDA (SourceCardOrWindow DestinationCard)                            (* Gaska "29-Aug-88 07:40")

        (* * Not entirely understood. (Mostly imported from TEDIT card.))

    (LET* ([Window (COND
                     ((WINDOWP SourceCardOrWindow)
                      SourceCardOrWindow)
                     (T (NC.FetchWindow SourceCardOrWindow]
           (Substance (NC.FetchSubstance (NC.CoerceToCard Window)))
           (ListOfNoteCardLinks (for Link&Position in (CDR Substance) collect (CAR Link&Position)))
           (SourceCard (NC.CoerceToCard Window))
           (WasDirtyFlg (NC.CardDirtyP SourceCard)))
          [for Link in ListOfNoteCardLinks when (NC.SameCardP (fetch (Link DestinationCard) of Link)
                                                              DestinationCard)
             do (NC.BitmapCardDisplayLinkIcon [CDR (ASSOC Link (WINDOWPROP Window (QUOTE LinkIcons]
                        Window
                        (CDR (ASSOC Link (CDR Substance]
          (NC.MarkCardDirty SourceCard (NOT WasDirtyFlg])
```

(NC.BitmapCardTranslateWindowPosition
```
  [LAMBDA (Card Window WindowPositionX WindowPositionY)                   (* Newman "22-Apr-87 17:57")
```

(* * Translate positions between window positions and bitmap positions.
Simple since the window system arranges everything for us.)

```
    (CONS WindowPositionX WindowPositionY])

)
```

(* * Auxiliary Functions)

```
(DEFINEQ
```

(NC.BitmapCardButtonEventFn
  [LAMBDA (Window)   (* Gaska "18-Aug-88 13:53")

         (* * BUTTONEVENTFN for bitmap cards.)

```
    (TOTOPW Window)
    (LET ((LinkAndRegion (NC.BitmapCardLinkFromPosition Window (LASTMOUSEPOSITION Window)))
          (SaveRegion (DSPCLIPPINGREGION NIL Window)))
         (if LinkAndRegion
             then (RESETLST
                        (RESETSAVE (DSPCLIPPINGREGION (CADR LinkAndRegion)
                                           Window)
                              (LIST (QUOTE DSPCLIPPINGREGION)
                                    SaveRegion Window))
                        (NC.LinkIconButtonEventInFn [CDR (ASSOC (CAR LinkAndRegion)
                                                         (WINDOWPROP Window (QUOTE LinkIcons]
                              Window))])
```

(NC.BitmapCardCopyButtonEventFn
  [LAMBDA (Window)   (* Gaska "15-Aug-88 15:55")

         (* * The sihft-select function for bitmap cards. Returns a bitmap imageobject.)

```
    (DECLARE (GLOBALVARS BITMAPIMAGEFNS))
    (INVERTW Window)
    (COPYINSERT (IMAGEOBJCREATE (LIST (CAR (NC.FetchSubstance (NC.CoerceToCard Window)))
                                      1 0)
                      BITMAPIMAGEFNS))
    (INVERTW Window])
```

(NC.BitmapCardDisplayLinkIcon
  [LAMBDA (Icon Window Position)   (* Gaska "29-Aug-88 07:27")

         (* * Display a link icon in a bitmap card.)

```
    (DSPXPOSITION (CAR Position)
          (WINDOWPROP Window (QUOTE DSP)))
    (DSPYPOSITION (CDR Position)
          (WINDOWPROP Window (QUOTE DSP)))
    (NC.LinkIconDisplayFn Icon Window])
```

(NC.BitmapCardShrinkFn
  [LAMBDA (Window)   (* Gaska "29-Aug-88 07:21")

         (* * Shrink Window to a bitmap card icon.)
                                                              (* update the link icon title.)
```
    [if (WINDOWPROP Window (QUOTE ICONWINDOW))
        then (ICONW.TITLE (WINDOWPROP Window (QUOTE ICONWINDOW))
                  (CONCAT "NC: " (NC.CardTitle (NCP.WhichCard Window]
    (OR (WINDOWPROP Window (QUOTE ICON))
        (LET [(NewIcon (TITLEDICONW (create TITLEDICON
                                            ICON _ NC.BitmapCardIconBitmap
                                            MASK _ NIL
                                            TITLEREG _
                                             (create REGION
                                                    LEFT _ 10
                                                    BOTTOM _ 10
                                                    WIDTH _ 80
                                                    HEIGHT _ 80))
                                (CONCAT "NC: " (NCP.CardTitle (NCP.WhichCard Window]
              (WINDOWPROP Window (QUOTE ICON)
                    NewIcon)
              NewIcon])
```

(NC.BitmapCardShowAllLinks
  [LAMBDA (Window)   (* Newman "22-Apr-87 18:05")

         (* * Show all thie links that should be shown in Window.)

```
    (bind (Substance _ (NCP.CardSubstance (NCP.WhichCard Window))) for Link&Icon in (WINDOWPROP Window
                                                                           (QUOTE LinkIcons))
```

```
        do (NC.BitmapCardDisplayLinkIcon (CDR Link&Icon)
                    Window
                    (CDR (ASSOC (CAR Link&Icon)
                                (CDR Substance])
```

(**NC.BitmapCardWindowSetup**
```
  [LAMBDA (Window Type Substance)                                      (* Gaska " 7-Sep-88 12:43")
```

        (* * Sets up a bitmap card window. Called by the make card fn, and the bring up card fn.)

        (* * Standard Window properties.)

```
    (WINDOWPROP Window (QUOTE ICONFN)
          (FUNCTION NC.BitmapCardShrinkFn))
    (WINDOWPROP Window (QUOTE COPYBUTTONEVENTFN)
          (FUNCTION NC.BitmapCardCopyButtonEventFn))
    (WINDOWPROP Window (QUOTE BUTTONEVENTFN)
          (FUNCTION NC.BitmapCardButtonEventFn))
    (WINDOWADDPROP Window (QUOTE CLOSEFN)
          (QUOTE NC.QuitCard))
    (WINDOWADDPROP Window (QUOTE REPAINTFN)
          (QUOTE NC.BitmapCardShowAllLinks))
```

        (* * Special Window properties.)

```
    [WINDOWPROP Window (QUOTE LinkIcons)
          (for Link&Position in (CDR Substance) collect (CONS (CAR Link&Position)
                                                              (NC.MakeLinkIcon (CAR Link&Position]
```

        (* * NoteCards button event functions.)

```
    (NC.InstallTitleBarLeftMenu Window Type)
    (NC.InstallTitleBarMiddleMenu Window Type)
    (NC.InstallTitleBarButtonEventFn Window (FUNCTION NC.TitleBarButtonEventFn))
    Window])
```

)

        (* * Bitmap editing functions specific to the Bitmap card)

(DEFINEQ

(**NC.BitmapAddLink**
```
  [LAMBDA (Window)                                                     (* Gaska "18-Aug-88 08:59")
```

        (* * Add a link to the bitmap in Window.)

```
    (NC.AddLinkToCard (NC.CoerceToCard Window])
```

(**NC.BitmapDeleteLink**
```
  [LAMBDA (Window)                                                     (* Gaska "28-Aug-88 11:24")
```

        (* * Move a Link in the Bitmap card represented by Window.)

```
    (RESETLST
        (RESETSAVE NIL (LIST (QUOTE CLOSEPROMPTWINDOW)
                             Window))
        (NCP.PrintMsg Window T "Select the link to be deleted")
        (LET [(Card (NC.CoerceToCard Window))
              (Link (CAR (NC.BitmapCardSelectLink Window]
             (if Link
                 then (NC.DeleteLink Link)
                      (NC.MarkCardDirty Card)
               else (NCP.PrintMsg Window T "No Link Selected.")
                    (DISMISS 2000))
             Window))])
```

(**NC.BitmapMoveLink**
```
  [LAMBDA (Window)                                                     (* Gaska "26-Sep-88 13:30")
```

        (* * Move a Link in the Bitmap card represented by Window.)

```
    (RESETLST
        (RESETSAVE NIL (LIST (QUOTE CLOSEPROMPTWINDOW)
                             Window))
        (NCP.PrintMsg Window T "Select the link to be moved")
        (LET* [(Card (NC.CoerceToCard Window))
               (Link (NC.BitmapCardSelectLink Window))
               (Position (if Link
                             then (NCP.PrintMsg Window T "Place the link in its new position")
                                  (GETBOXPOSITION (fetch WIDTH of (CADR Link))
                                      (fetch HEIGHT of (CADR Link))
                                      NIL NIL Window]
```

```
                    (IF Link
                       THEN (if (INSIDEP (DSPCLIPPINGREGION NIL Window)
                                         Position)
                                then (RPLACD (ASSOC (CAR Link)
                                                    (CDR (NC.FetchSubstance Card)))
                                             Position)
                                     (REDISPLAYW Window)
                                     (NC.MarkCardDirty Card)
                              else (NCP.PrintMsg Window T "Position outside window.")
                                   (DISMISS 2000))
                       ELSE (NCP.PrintMsg Window T "No Link selected.")
                            (DISMISS 2000))
                    (CLOSEPROMPTWINDOW Window)
                    Window))])
)
```

(* * Auxiliary editing functions.)

(* * Temporary function until patch appears.)

(DEFINEQ

(**NC.AddLinkToCard**
```
  [LAMBDA (CardIdentifier LinkOrLinkLabel DestinationCard)          (* Newman "20-Jul-88 16:15")
```

(* * Add a link to card designated by CardIdentifier. Rest of args can be NIL.
Currently, this dispatches on card type. Should eventually call something like an InsertLinkFn.)

(* * fgh 6/9/86 Added code to check to make sure that another operation is not in progress on this card when this fn is
called.)

(* * rg 3/4/87 rewritten for new version of NC.ProtectedCardOperation)

(* * dvn attempt to dispatch on card type.)

```
    (LET ((Card (NC.CoerceToCard CardIdentifier))
          CardType Window)
         (NC.ProtectedCardOperation Card "Insert Link" NIL (SETQ Window (NC.FetchWindow Card))
               (SETQ CardType (NC.FetchType Card))
               (OR LinkOrLinkLabel (SETQ LinkOrLinkLabel (NC.AskLinkLabel Window NIL NIL T T)))
               (AND LinkOrLinkLabel (COND
                                      ((NC.IsSubTypeOfP CardType (QUOTE Text))
                                       (NC.InsertLinkInText (NC.FetchSubstance Card)
                                             LinkOrLinkLabel DestinationCard Card))
                                      ((NC.IsSubTypeOfP CardType (QUOTE Graph))
                                       (NC.AddLinkToGraphCard Window LinkOrLinkLabel DestinationCard))
                                      ((NC.IsSubTypeOfP CardType (QUOTE Sketch))
                                       (NC.AddLinkToSketchCard Window LinkOrLinkLabel DestinationCard))
                                      ((NCP.CardTypeFn CardType (QUOTE InsertLinkFn))
                                       (APPLY* (NCP.CardTypeFn CardType (QUOTE InsertLinkFn))
                                             Card Window LinkOrLinkLabel NIL DestinationCard))
                                      (T (NCP.ReportError (QUOTE NC.AddLinkToCard)
                                                (CONCAT "Cannot insert link in card type " CardType]
)
```

(* * Card Type Creation)

(DEFINEQ

(**NC.AddBitmapCardType**
```
  [LAMBDA NIL                                                        (* Gaska "26-Sep-88 11:13")
```

(* * Define a new card type called the bitmap card. It does not inherit from anything but the base type.)

```
    (NCP.CreateCardType (QUOTE Bitmap)
          (QUOTE NoteCard)
          (QUOTE ((MakeFn NC.BitmapCardCreate)
                  (EditFn NC.BitmapCardEdit)
                  (QuitFn NC.BitmapCardCloseFn)
                  (GetFn NC.BitmapCardGetSubstance)
                  (PutFn NC.BitmapCardPutSubstance)
                  (CopyFn NC.BitmapCardCopySubstance)
                  (MarkDirtyFn NC.BitmapCardMarkDirty)
                  (DirtyPFn NC.BitmapCardDirtyP)
                  (CollectLinksFn NC.BitmapCardCollectLinks)
                  (DeleteLinksFn NC.BitmapCardDeleteLinks)
                  (UpdateLinkIconsFn NC.BitmapCardUpdateLinks)
                  (InsertLinkFn NC.BitmapCardInsertLinks)
                  (TranslateWindowPositionFn NC.BitmapCardTranslateWindowPosition)))
          (BQUOTE ((LinkDisplayMode Title)
                   (LinkAnchorModesSupported T)
                   (LinkIconAttachedBitMap , NC.BitmapCardBitmap)
                   (DisplayedInMenuFlg T])
```

)

      (* * Auxiliary files)

```
(FILESLOAD NCTYPESMECH (SYSLOAD FROM LISPUSERS)
        IMAGEOBJ)

(FILESLOAD BITMAPEDITOR)
```

      (* * Variables)

```
(RPAQ NC.BitmapCardBitmap (READBITMAP))
```

```
(20 17 "OOOOO@@@" "H@@@A@@@" "H@@@A@@@" "H@@@A@@@" "HA@BA@@@" "HFLBA@@@" "HHBBA@@@" "HHBBA@@@" "I@ABA@@@"
    "HHBBA@@@" "HHBBA@@@" "HFLBA@@@" "HA@BA@@@" "H@@@A@@@" "H@@@A@@@" "H@@@A@@@" "OOOOO@@@")
```

```
(RPAQ NC.BitmapCardIconBitmap (READBITMAP))
```

```
(100 100 "OOOOOOOOOOOOOOOOOOOOOOOOOOO@@@" "OOOOOOOOOOOOOOOOOOOOOOOOOOO@@@" "O@NEJHHMHK@ECNCCGI@ALALHK@@@"
    "NC@ELGGONB@OAHAEBDAIDAMHK@@@" "MA@AOJO@LC@OALCBKMBFDFAGO@@@" "LAI@IM@LBKHNICEGFHHHKBDCK@@@"
    "LAM@OLOHIHGGGDEHNKHGKJDOG@@@" "MNHOHEOFIFGGEJBJJHIOKDL@C@@@" "MNADIINJIFHBFG@@DGFDOM@NC@@@"
    "ONOJA@AHIAHALGKOBAAGLKONG@@@" "LFL@@@@@@@@@@@@@@@@@@BKC@@@" "NKL@@@@@@@@@@@@@@@@@@@AFK@@@"
    "MJH@@@@@@@@@@@@@@@@@@@@OC@@@" "LED@@@@@@@@@@@@@@@@@@@BAO@@@" "NND@@@@@@@@@@@@@@@@@@@CFG@@@"
    "MLL@@@@@@@@@@@@@@@@@@@MO@@@" "LJ@@@@@@@@@@@@@@@@@@@@JC@@@" "MIL@@@@@@@@@@@@@@@@@@@CCO@@@"
    "LKD@@@@@@@@@@@@@@@@@@BFG@@@" "OEH@@@@@@@@@@@@@@@@@@@KC@@@" "MHL@@@@@@@@@@@@@@@@@@CMO@@@"
    "MNL@@@@@@@@@@@@@@@@@@BCO@@@" "OLH@@@@@@@@@@@@@@@@@C@G@@@" "MKD@@@@@@@@@@@@@@@@@@ADG@@@"
    "OF@@@@@@@@@@@@@@@@@@@CJC@@@" "NC@@@@@@@@@@@@@@@@@@@BHC@@@" "NJH@@@@@@@@@@@@@@@@@@OO@@@"
    "L@D@@@@@@@@@@@@@@@@@@CIG@@@" "NAD@@@@@@@@@@@@@@@@@@GO@@@" "NAD@@@@@@@@@@@@@@@@@@@C@@@"
    "NFH@@@@@@@@@@@@@@@@@@AHC@@@" "NCD@@@@@@@@@@@@@@@@@@AKO@@@" "NL@@@@@@@@@@@@@@@@@@@AAC@@@"
    "NB@@@@@@@@@@@@@@@@@@@BGC@@@" "MOH@@@@@@@@@@@@@@@@@@AFK@@@" "MJL@@@@@@@@@@@@@@@@@@@LC@@@"
    "LE@@@@@@@@@@@@@@@@@@@CHG@@@" "NDD@@@@@@@@@@@@@@@@@@AGO@@@" "ONL@@@@@@@@@@@@@@@@@@CLK@@@"
    "LOL@@@@@@@@@@@@@@@@@@BAK@@@" "LMD@@@@@@@@@@@@@@@@@@AGC@@@" "OOD@@@@@@@@@@@@@@@@@@CMK@@@"
    "NE@@@@@@@@@@@@@@@@@@@BOG@@@" "OH@@@@@@@@@@@@@@@@@@@BNK@@@" "NJD@@@@@@@@@@@@@@@@@@GK@@@"
    "NLH@@@@@@@@@@@@@@@@@@ANC@@@" "N@H@@@@@@@@@@@@@@@@@@FK@@@" "LN@@@@@@@@@@@@@@@@@@@FK@@@"
    "MND@@@@@@@@@@@@@@@@@@ANG@@@" "MHD@@@@@@@@@@@@@@@@@@KO@@@" "NHL@@@@@@@@@@@@@@@@@@AGK@@@"
    "OM@@@@@@@@@@@@@@@@@@@AHO@@@" "NNH@@@@@@@@@@@@@@@@@@CC@@@" "MB@@@@@@@@@@@@@@@@@@@HO@@@"
    "NJD@@@@@@@@@@@@@@@@@@NO@@@" "NMH@@@@@@@@@@@@@@@@@@BC@@@" "MKD@@@@@@@@@@@@@@@@@@FG@@@"
    "MN@@@@@@@@@@@@@@@@@@@CO@@@" "MGL@@@@@@@@@@@@@@@@@@ALC@@@" "MA@@@@@@@@@@@@@@@@@@@DK@@@"
    "N@D@@@@@@@@@@@@@@@@@@AC@@@" "L@D@@@@@@@@@@@@@@@@@@BC@@@" "NBH@@@@@@@@@@@@@@@@@@@O@@@"
    "OM@@@@@@@@@@@@@@@@@@@ANC@@@" "OLD@@@@@@@@@@@@@@@@@@KC@@@" "OOL@@@@@@@@@@@@@@@@@@BCG@@@"
    "LED@@@@@@@@@@@@@@@@@@BCO@@@" "NJH@@@@@@@@@@@@@@@@@@KK@@@" "NHL@@@@@@@@@@@@@@@@@@DC@@@"
    "OIH@@@@@@@@@@@@@@@@@@BJC@@@" "OD@@@@@@@@@@@@@@@@@@@CIG@@@" "LAD@@@@@@@@@@@@@@@@@@COC@@@"
    "LD@@@@@@@@@@@@@@@@@@@BGO@@@" "LD@@@@@@@@@@@@@@@@@@@BJK@@@" "LA@@@@@@@@@@@@@@@@@@@BAG@@@"
    "LGD@@@@@@@@@@@@@@@@@@BCG@@@" "NLL@@@@@@@@@@@@@@@@@@BMG@@@" "LLD@@@@@@@@@@@@@@@@@@AGC@@@"
    "OBH@@@@@@@@@@@@@@@@@@CLO@@@" "OAL@@@@@@@@@@@@@@@@@@CGK@@@" "MND@@@@@@@@@@@@@@@@@@FK@@@"
    "OEL@@@@@@@@@@@@@@@@@@CBG@@@" "MHD@@@@@@@@@@@@@@@@@@ALG@@@" "OAH@@@@@@@@@@@@@@@@@@CHO@@@"
    "OED@@@@@@@@@@@@@@@@@@GC@@@" "OGL@@@@@@@@@@@@@@@@@@CHG@@@" "NAL@@@@@@@@@@@@@@@@@@BAG@@@"
    "LOL@@@@@@@@@@@@@@@@@@BJC@@@" "LD@@@@@@@@@@@@@@@@@@@CDC@@@" "LMD@@@@@@@@@@@@@@@@@@CKK@@@"
    "NMEMMJGON@L@NAH@@OOEBO@MK@@@" "LMMOBFOAMDHJGGINLOJAG@DCK@@@" "NKJDJJOAHEJMK@KCBLFIACOBO@@@"
    "OBNJIEBCHEGCGDEOO@HKEBNEC@@@" "ODM@CCCNNCF@EFGELHDHIGDIO@@@" "ODCDLGCKMFOJFKHLBADMLNLAC@@@"
    "MLEGG@GAMBOJLJCCNCANBG@AC@@@" "OIOFONIOMFGGE@BJIICABDHOC@@@" "OOOOOOOOOOOOOOOOOOOOOOOOOOO@@@"
    "OOOOOOOOOOOOOOOOOOOOOOOOOOO@@@")

(RPAQQ NC.BitmapCardMenuItems [("Links" (NC.BitmapAddLink)
                                        "Allow the user to manipulate links in the bitmap."
                                        (SUBITEMS ("Add Link" (NC.BitmapAddLink))
                                                ("Delete Link" (NC.BitmapDeleteLink))
                                                ("Move Link" (NC.BitmapMoveLink)])

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.BitmapCardBitmap NC.BitmapCardIconBitmap NC.BitmapCardMenuItems)
)
```

      (* * Initialization)

```
(NC.AddBitmapCardType)

(PUTPROPS NCBITMAPCARD COPYRIGHT ("Xerox Corporation" 1987 1988))
```

FUNCTION INDEX

VARIABLE INDEX