

File created: 8-Mar-2024 15:58:49 {DSK}<home>frank<il>notecards>system>NCLINKS.;5

changes to: (FNS NC.InsertLinkInText)

previous date: 8-Mar-2024 12:40:39 {DSK}<home>frank<il>notecards>system>NCLINKS.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ NCLINKSCOMS

(

;;; Internal variables

```
(GLOBALVARS NC.LinkIconSpaceInterLine NC.FileBoxLinkSpacer NC.NoDeleteImageFns NC.SystemLinkLabels
NC.InitialLinkLabels NC.SubBoxLinkLabel NC.FileCardLinkLabel NC.BrowserContentsLinkLabel
NC.ListContentsLinkLabel NC.UnspecifiedLinkLabel NC.UCASESystemLinkLabels NC.DocBackPtrLinkLabel
NC.LinkIndexBackPtrLinkLabel NC.LinkIconImageFns NC.LinkIconMiddleButtonMenu
NC.LinkDisplayModesStylesheet NC.LinkIconBorderWidth NC.LinkIconSpaceOuterY
NC.LinkIconSpaceInnerY NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerX NC.OrphanID
MINIMUM.VISIBLE.SCALE.FACTOR DEFAULT.VISIBLE.SCALE.FACTOR TEDIT.DEFAULT.FONT)
(INITVARS (NC.SubBoxLinkLabel 'SubBox)
(NC.FileCardLinkLabel 'FileCard)
(NC.BrowserContentsLinkLabel 'BrowserContents)
(NC.ListContentsLinkLabel 'ListContents)
(NC.UnspecifiedLinkLabel 'Unspecified)
(NC.DocBackPtrLinkLabel 'DocBackPtr)
(NC.LinkIndexBackPtrLinkLabel 'LinkIndexBackPtr)
(NC.SystemLinkLabels (LIST NC.ListContentsLinkLabel NC.BrowserContentsLinkLabel
NC.FileCardLinkLabel NC.SubBoxLinkLabel NC.DocBackPtrLinkLabel
NC.LinkIndexBackPtrLinkLabel))
(NC.UCASESystemLinkLabels (for Label in NC.SystemLinkLabels collect (U-CASE Label)))
(NC.InitialLinkLabels (APPEND '(Comment See Unspecified)
NC.SystemLinkLabels))
(NC.LinkIconSpaceInterLine 0)
(NC.LinkIconBorderWidth 1)
(NC.LinkIconSpaceOuterY 2)
(NC.LinkIconSpaceInnerY 1)
(NC.LinkIconSpaceOuterX 2)
(NC.LinkIconSpaceInnerX 3)
(NC.FileBoxLinkSpacer (CHARACTER 13)))
```

;;; Links mechanism

```
(FNS NC.GlobalLinkP NC.ChildLinkP NC.ContentsLinkP NC.SubContentsLinkP NC.ReverseLinkLabelP
NC.MakeGlobalLink NC.AddFromLink NC.AddLinkToGraphCard NC.AddLinkToSketchCard NC.InsertLinkInSketch
NC.InsertLinkInText NC.AddToLink NC.CachedAddToLink NC.DelFromLink NC.CachedDelFromLink
NC.DelToLink NC.DelReferencesToCard NC.DelReferencesToCardFromGlobalList NC.DeleteLink
NC.DeleteToLink NC.DeleteFromLink NC.DeletedLinkDisplayFn NC.DeletedLinkGetFn
NC.DeletedLinkImageBoxFn NC.DeletedLinkPutFn NC.DeletedLinkImageObjP NC.DeleteBadLink
NC.ReplaceWithDeletedLinkImageObj NC.CheckForOrphanDelete NC.FileInOrphanBox NC.HookToOrphanCard
NC.InsertLinkBeforeMarker NC.LinkLabelP NC.IDAlphaOrder NC.MakeLink NC.CachedMakeFilingLink
NC.MakeFilingLink NC.MakeFilingLinks NC.RelabelLink NC.SystemLinkLabelP NC.ValidLinkP
NC.MakeChildLink NC.EnsureNoCycles NC.SameLinkP NC.SameLinksP NC.AddGlobalLinkToCard
NC.AddGlobalLinksToCard NC.ChangeCardTitleFromLinkIcon NC.UncacheLinks NC.TraverseLink
NC.CreateLinkIconStrings NC.ShrinkIcon)
(FNS NC.AddLinkToCard NC.AddLinksToCard)
(FNS NC.DelReferencesToCardFromShowLinks NC.UpdateLinkImagesInShowLinks NC.FetchShowLinksWindow
NC.ChangeLinkLabelInShowLinksWin)
```

;;; New Display Mode format

```
(FNS NC.CheckDisplayModeFormat NC.InsureLinkDisplayMode NC.MakeNewDisplayMode
NC.DisplayModeFromStylesheetSelections NC.StylesheetSelectionsFromDisplayMode)
(GLOBALVARS NC.MenuItems/DisplayModeValues)
[INITVARS (NC.MenuItems/DisplayModeValues '(Yes T No NIL System% Defaults FLOAT Yes T No NIL
System% Defaults FLOAT Yes T No NIL System% Defaults
FLOAT))
```

;;; New fns to support No-Link cards/substances

```
(FNS NC.LinksSupportedP)
```

;;; Link display icons

```
(FNS NC.FetchLinkIconForLink NC.MakeLinkIcon NC.LinkIconButtonEventInFn NC.LinkIconDisplayFn
NC.LinkIconImageBoxFn NC.LinkIconImageObjP NC.LinkIconPutFn NC.LinkIconGetFn
NC.LinkIconMiddleButtonFn NC.LinkIconWhenCopiedFn NC.LinkIconWhenDeletedFn
NC.LinkIconWhenInsertedFn NC.LinkIconWhenMovedFn NC.FetchLinkFromLinkIcon NC.ChangeLinkDisplayMode
NC.UpdateLinkImages NC.LinkIconCopyFn NC.LinkAtCharPos NC.LinkIconLeftButtonFn NC.LinkToBitMap
NC.SetBitMap&Box)
```

(FNS NC.FillInLinkIcon NC.InvisibleLinkImageBoxFn NC.InvisibleLinkPutFn NC.InvisibleLinkGetFn)

;;; Pointer icons. These act like links, except that their icon boxes are dashed, and they're invisible as far as the source or destination cards are concerned. That is, only the imageobject containing the pointer knows about the connection. Functions exist for creating pointer icons and converting them into link icons.

```
(GLOBALVARS NC.PointerIconImageObjFns NC.PointerIconDashingStyle NC.PointerIconMiddleButtonMenu)
[INITVARS (NC.PointerIconImageObjFns (IMAGEFNSCREATE (FUNCTION NC.PointerIconDisplayFn)
              (FUNCTION NC.PointerIconImageBoxFn)
              (FUNCTION NC.PointerIconPutFn)
              (FUNCTION NC.PointerIconGetFn)
              (FUNCTION NC.PointerIconCopyFn)
              (FUNCTION NC.LinkIconButtonEventInFn)))
          (NC.PointerIconDashingStyle ' (4 2))
```

;;; Functions likely to be useful externally.

```
(FNS NC.InsertPointerInText NC.ConvertPointerIconToLinkIcon NC.MakePointerIcon NC.MakePointer
  NC.PointerIconImageObjP NC.FetchPointerFromPointerIcon)
```

;;; Internal functions.

```
(FNS NC.PointerIconDisplayFn NC.PointerIconImageBoxFn NC.PointerIconPutFn NC.PointerIconGetFn
  NC.PointerIconCopyFn NC.WritePointer NC.ReadPointer)
(FNS NC.PointerIconLeftButtonFn NC.PointerIconMiddleButtonFn NC.TraversePointer
  NC.ChangePointerDisplayMode NC.RelabelPointer NC.ChangeCardTitleFromPointerIcon)
(FNS NC.DeleteBadPointer NC.DrawLinkOrPointerIcon NC.ComputeLinkOrPointerImageBox)
(DECLARE%: DONTEVAL@LOAD (P (NC.MakePointerIcon)))
```

;;; New stuff for allowing put to work from TEdit.

```
(GLOBALVARS NC.ExternalPutLinkIconImageFns)
[INITVARS (NC.ExternalPutLinkIconImageFns (IMAGEFNSCREATE (FUNCTION NC.ExternalPutLinkIconDisplayFn)
              (FUNCTION NC.ExternalPutLinkIconImageBoxFn)
              (FUNCTION NC.ExternalPutLinkIconPutFn)
              (FUNCTION NC.ExternalPutLinkIconGetFn)
              NIL
              (FUNCTION NIL))
          (FNS NC.ExternalPutLinkIconDisplayFn NC.ExternalPutLinkIconImageBoxFn NC.ExternalPutLinkIconPutFn
            NC.ExternalPutLinkIconGetFn NC.ExternalGetLinkIconFn NC.CoerceToExternalPutLinkIcon)
```

;;; Stuff to register our image object for intermezzo.

```
(FNS NC.RegisterPlaceMarkerImageObj)
(DECLARE%: DONTEVAL@LOAD (P (NC.RegisterPlaceMarkerImageObj)))
```

;;; Stuff for Having BitMap external Link Images

```
(GLOBALVARS NC.ExternalBitMapPutLinkIconImageFns NC.LinkIconBitMapStream)
[INITVARS (NC.ExternalBitMapPutLinkIconImageFns (IMAGEFNSCREATE (FUNCTION
              NC.ExternalPutLinkIconDisplayFn)
              (FUNCTION NC.ExternalPutLinkIconImageBoxFn)
              (FUNCTION NC.ExternalPutLinkIconPutFn)
              (FUNCTION NC.ExternalGetLinkIconFn)
              NIL
              (FUNCTION NIL))
          [P (SETQ NC.LinkIconBitMapStream (DSPCREATE (BITMAPCREATE 512 128))
```

;;; Stuff for handling link icons in sketches.

```
(FNS NC.LinkIconSketchElementP NC.DeleteLinkIconSketchElement)
```

;;; Icons representing notecards types.

```
(GLOBALVARS NC.FileBoxIcon NC.GraphCardIcon NC.SketchCardIcon NC.TextCardIcon NC.TypelessIcon)
```

;;; DELETEME imageobjs

```
(GLOBALVARS NC.DELETEMEImageObjDatum NC.RemoveDELETEMEImageObjsFromCardFlg)
(INITVARS (NC.DELETEMEImageObjDatum "[[DELETE ME]]")
          (NC.RemoveDELETEMEImageObjsFromCardFlg NIL))
(FNS NC.DELETEMEImageObjP)
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
      NCLINKS))
```

;;; Internal variables

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.LinkIconSpaceInterLine NC.FileBoxLinkSpacer NC.NoDeleteImageFns NC.SystemLinkLabels
NC.InitialLinkLabels NC.SubBoxLinkLabel NC.FiledCardLinkLabel NC.BrowserContentsLinkLabel
NC.ListContentsLinkLabel NC.UnspecifiedLinkLabel NC.UCASESystemLinkLabels NC.DocBackPtrLinkLabel
NC.LinkIndexBackPtrLinkLabel NC.LinkIconImageFns NC.LinkIconMiddleButtonMenu
NC.LinkDisplayModesStylesheet NC.LinkIconBorderWidth NC.LinkIconSpaceOuterY NC.LinkIconSpaceInnerY
NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerX NC.OrphanID MINIMUM.VISIBLE.SCALE.FACTOR
DEFAULT.VISIBLE.SCALE.FACTOR TEDIT.DEFAULT.FONT)
)

(RPAQ? NC.SubBoxLinkLabel 'SubBox)

(RPAQ? NC.FiledCardLinkLabel 'FiledCard)

(RPAQ? NC.BrowserContentsLinkLabel 'BrowserContents)

(RPAQ? NC.ListContentsLinkLabel 'ListContents)

(RPAQ? NC.UnspecifiedLinkLabel 'Unspecified)

(RPAQ? NC.DocBackPtrLinkLabel 'DocBackPtr)

(RPAQ? NC.LinkIndexBackPtrLinkLabel 'LinkIndexBackPtr)

(RPAQ? NC.SystemLinkLabels (LIST NC.ListContentsLinkLabel NC.BrowserContentsLinkLabel NC.FiledCardLinkLabel
NC.SubBoxLinkLabel NC.DocBackPtrLinkLabel NC.LinkIndexBackPtrLinkLabel))

(RPAQ? NC.UCASESystemLinkLabels (for Label in NC.SystemLinkLabels collect (U-CASE Label)))

(RPAQ? NC.InitialLinkLabels (APPEND '(Comment See Unspecified)
NC.SystemLinkLabels))

(RPAQ? NC.LinkIconSpaceInterLine 0)

(RPAQ? NC.LinkIconBorderWidth 1)

(RPAQ? NC.LinkIconSpaceOuterY 2)

(RPAQ? NC.LinkIconSpaceInnerY 1)

(RPAQ? NC.LinkIconSpaceOuterX 2)

(RPAQ? NC.LinkIconSpaceInnerX 3)

(RPAQ? NC.FileBoxLinkSpacer (CHARACTER 13))

::: Links mechanism

(DEFINEQ

(NC.GlobalLinkP

[LAMBDA (Link)

(\* kirk%: "15-Nov-85 03:36")

(\* \* Return T if Link is a Global to Global link)

(EQ 'GlobalGlobal (fetch (Link AnchorMode) of Link))

(NC.ChildLinkP

[LAMBDA (Link)

(\* rht%: " 1-Aug-84 09:34")

(\* \* Is link either to a filed card or sub filebox?)

(OR (NC.ContentsLinkP Link)
(NC.SubContentsLinkP Link))

(NC.ContentsLinkP

[LAMBDA (Link)

(\* kirk%: "15-Nov-85 01:42")

(COND
((EQ NC.FiledCardLinkLabel (fetch (Link Label) of Link))
T)
(T NIL))

(NC.SubContentsLinkP

[LAMBDA (Link)

(\* kirk%: "15-Nov-85 01:42")

(COND
((EQ NC.SubBoxLinkLabel (fetch (Link Label) of Link))
T)
(T NIL))

(NC.ReverseLinkLabelP

[LAMBDA (Link ListOfLabels)

(\* kirk%: "15-Nov-85 01:43")

(\* Is the reverse of Link, i.e. with "\_" prefix attached, present on

ListOfLabels?)

(\* rht 9/4/84%: Modified so as to return the linklabel found or nil.)

```
(PROG ((Label (fetch (Link Label) of Link)))
  (RETURN (COND
    ((FMEMB (PACK* '_ Label)
      (OR (LISTP ListOfLabels)
        (LIST ListOfLabels)))
      Label]))
```

**(NC.MakeGlobalLink**

[LAMBDA (Window LinkLabel DestinationCard SourceCard DisplayMode) (\* rht%: "26-Oct-87 19:32")

(\* Make a new global link from DestinationCard to SourceCard with label LinkLabel and display mode DisplayMode. Window is optional if SourceCard is specified, otherwise it is required.)

(\* rht 11/9/84%: Now checks result of NC.MakeLink before building link.)

(\* rht 10/26/87%: Now lets functions under NC.MakeLink take care of adding to global links list.)

```
(NC.MakeLink Window LinkLabel DestinationCard SourceCard DisplayMode 'GlobalGlobal NIL NIL 'ATEND])
```

**(NC.AddFromLink**

[LAMBDA (Link) (\* Randy.Gobbel "15-Jul-87 17:54")

(\* Add Link to the FromLinks of the appropriate card on DatabaseStream)

(\* kirk%: 14Nov85%: deleted use of DatabaseStream)

(\* rht 2/14/86%: Now calls NC.UncacheLinks instead of two calls to NC.SetToLinks and NC.SetFromLinks.)

(\* kef 7/22/86%: Now only calls NC.PutFromLinks instead of NC.PutLinks because we can't be certain of owning the write locks on all of the Link parts.)

(\* rg [5/15/87] now calls NC.LinksCachedP instead of NC.ActiveCardP)

```
(LET ((Card (fetch (Link DestinationCard) of Link)))
  [COND
    ((NC.LinksCachedP Card)
      (NC.SetFromLinks Card (CONS Link (NC.FetchFromLinks Card)))
      (NC.SetLinksDirtyFlg Card T))
    (T (WITH-MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of Card))
      (NC.GetLinks Card)
      (NC.SetFromLinks Card (CONS Link (NC.FetchFromLinks Card)))
      (NC.PutFromLinks Card)
      (NC.UncacheLinks Card)))]
  Link])
```

**(NC.AddLinkToGraphCard**

[LAMBDA (Window LinkOrLinkLabel DestinationCard) ; Edited 1-Jun-88 17:47 by pmi

;; Add a NOTECARD link to a graph card. Pass the fact that this is a link-add to NC.GraphAddNodeFn via the NoteCardInsertingLink window  
;; property

;; rht 4/7/86: Now can handle insertion of multiple links by accepting label and/or destination card args. Either or both can be nil.

;; rht 11/2/86: Now checks result of NC.MakeLink before proceeding.

;; pmi 6/1/88: Now makes sure that GRAPH.ADDNODEFN of this Window's graph is NC.GraphAddNodeFn; it may have been changed elsewhere.

```
(LET [(AttachW (NC.AttachPromptWindow Window))
  (Link (if (type? Link LinkOrLinkLabel)
    then LinkOrLinkLabel
    else (NC.MakeLink Window LinkOrLinkLabel DestinationCard]
  (if Link
    then (WINDOWPROP Window 'NoteCardInsertingLink Link)
    (replace (GRAPH GRAPH.ADDNODEFN) of (WINDOWPROP Window 'GRAPH) with (FUNCTION NC.GraphAddNodeFn))
    (EDITADDNODE Window NIL AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
    (CLOSEW AttachW]))
```

**(NC.AddLinkToSketchCard**

[LAMBDA (SketchWindow LinkOrLinkLabel DestinationCard) (\* rht%: " 7-Apr-86 17:24")

(\* Called from a SKETCH window to add a new NoteCard Link Icon to the sketch)

(\* rht 11/9/84%: Now checks result of NC.MakeLink before building link.)

(\* rht 4/7/86%: Now accepts LinkOrLinkLabel and DestinationCard args. Either or both can be nil.)

```
(if (NOT (type? Link LinkOrLinkLabel))
  then (SETQ LinkOrLinkLabel (NC.MakeLink SketchWindow (OR LinkOrLinkLabel (NC.AskLinkLabel SketchWindow
    NIL NIL T T))
```

```

DestinationCard)))
(NC.InsertLinkInSketch SketchWindow LinkOrLinkLabel])

```

**(NC.InsertLinkInSketch**

```

[LAMBDA (SketchWindow Link SketchWindowPosition) (* kirk%: " 4-Sep-86 18:42")

```

```

(** rht 8/20/85%: Rewritten to use Richard's sketch programmer's interface.
Eliminates references to sketch records.)

(** rht 10/1/85%: Now checks that chosen position is actually within sketch, otherwise aborts by deleting link.)

(** fgh |2/6/86| Added Card argument.)

(** kirk 11Mar86 Deleted Card argument.)

(** rht 8/1/86%: Now doesn't break if NULL Link.)

(** kirk |9/4/86| changed to use VIEWER.SCALE instead of old name)

```

```

(AND Link (LET ((LinkIcon (NC.MakeLinkIcon Link))
                (Scale (VIEWER.SCALE SketchWindow))
                SketchIcon Size Position)
              (SETQ Size (IMAGEBOXSIZE LinkIcon))
              (SETQ Position (OR (POSITIONP SketchWindowPosition)
                                (GETBOXPOSITION (fetch (IMAGEBOX XSIZE) of Size)
                                                  (fetch (IMAGEBOX YSIZE) of Size)
                                                  NIL NIL SketchWindow))))
            (if (AND (INSIDEP (DSPCLIPPINGREGION NIL SketchWindow)
                          Position)
                  (SETQ SketchIcon (SKETCH.IMAGE.OBJECT.ELEMENT LinkIcon Scale Position)))
                then (SKETCH.ADD.ELEMENT SketchIcon SketchWindow)
                  SketchIcon
                else (NCP.DeleteLinks Link)
                    NIL))

```

**(NC.InsertLinkInText**

```

[LAMBDA (TextStream LinkOrLinkLabel DestinationCard SourceCard DisplayMode CharacterPosition)
(* kirk%: "15-Nov-85 11:53")

```

```

(** Add a link image object (and hence a link) to the text specified by TextStream.
Link has LinkLabel, goes to DestinationCard and has DisplayMode.
Returns the Link inserted.)

(** rht 11/9/84%: Now checks result of NC.MakeLink before building link.)

(** rht 3/26/85%: Now uses insert position to calculate the link after which to insert new link and passes this to
NC.MakeLink.)

```

```

(PROG ((Window (NC.TEditWindow TextStream T))
      Link InsertPos)
      (RETURN (COND
                ((SETQ InsertPos (OR (if (AND (FIXP CharacterPosition)
                                             (ILEQ CharacterPosition (TEDIT.NCHARS TextStream)))
                                       then CharacterPosition)
                                     (NC.CharPosFromTextStream TextStream)))
              (if [SETQ Link (COND
                    ((type? Link LinkOrLinkLabel)
                     LinkOrLinkLabel)
                    (T (NC.MakeLink Window LinkOrLinkLabel DestinationCard SourceCard
                                     DisplayMode NIL NIL NIL (NC.LinkAtCharPos InsertPos TextStream)]
                    then (TEDIT.SETSEL TextStream InsertPos 0 'LEFT)
                        (TEDIT.INSERT.OBJECT (NC.MakeLinkIcon Link)
                                             TextStream InsertPos))
                    Link)
                  (T (NC.ReportError "NC.InsertLinkInText" "No selection set when inserting link image
object.")
                    NIL))

```

**(NC.AddToLink**

```

[LAMBDA (Link LinkToInsertAfter BeingMovedFlg) ; Edited 9-Dec-88 18:46 by krivacic

```

```

;;; Add Link to the ToLinks of the source card

;;; rht 3/26/85: Added LinkToInsertAfter arg which is either NIL, ATEND, or a link after which to insert new link. If NIL, then insert at front of list. If
;;; ATEND then insert at end.

;;; kirk: 13Nov85: deleted second parameter (DatabaseStream) and changed to call NC.CachedAddToLink.

;;; rht 2/14/86: Now calls NC.UncacheLinks instead of two calls to NC.SetToLinks and NC.SetFromLinks.

;;; rht 8/11/86: Added BeingMovedFlg. If non-nil, then delete instance of Link from ToLinks before inserting it in new spot.

```

```

[LET ((SourceCard (fetch (Link SourceCard) of Link)))
(COND

```

```

((NC.ActiveCardP SourceCard)
 (NC.CachedAddToLink Link SourceCard LinkToInsertAfter BeingMovedFlg))
(T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
   (NC.GetLinks SourceCard)
   (NC.CachedAddToLink Link SourceCard LinkToInsertAfter BeingMovedFlg)
   (NC.PutLinks SourceCard)
   (NC.UncacheLinks SourceCard)))
Link])

```

**(NC.CachedAddToLink**

```

[LAMBDA (Link SourceCard LinkToInsertAfter BeingMovedFlg) (* rht%: "26-Oct-87 19:31")

  (** Add Link to the cached ToLinks of the source card)

  (** rht 3/26/85%: Added LinkToInsertAfter arg which is either NIL, ATEND, or a link after which to insert new link.
  If NIL, then insert at front of list. If ATEND then insert at end.)

  (** kirk%: 13Nov85%: changed to use NoteFile instead of DatabaseStream)

  (** kirk 15Nov85%: deleted use of LinkID)

  (** rht 8/11/86%: Added BeingMovedFlg. If non-nil, then delete instance of Link from ToLinks before inserting it in new spot.)

  (** rht 10/26/87%: Now adds to global links list if necessary.)

(LET ((ToLinks (NC.FetchToLinks SourceCard)))
 [if BeingMovedFlg
  then
    (if (NC.SameLinkP Link (CAR ToLinks)) (* Look for and delete old instance of the link.)
      then (NC.SetToLinks SourceCard (SETQ ToLinks (CDR ToLinks)))
      else (for RestOfToLinks on ToLinks thereis (NC.SameLinkP Link (CADR RestOfToLinks))
            finally (RPLACD RestOfToLinks (CDDR RestOfToLinks))
          )
    )
 [COND
  [(AND ToLinks (type? Link LinkToInsertAfter)) (* Destructively insert after a given link.)
   (for RestOfToLinks on ToLinks thereis (NC.SameLinkP LinkToInsertAfter (CAR RestOfToLinks))
    finally (COND
      [RestOfToLinks (RPLACD RestOfToLinks (CONS Link (CDR RestOfToLinks))
        (T (NC.SetToLinks SourceCard (NCONC1 ToLinks Link)
          ((EQ LinkToInsertAfter 'ATEND) (* Destructively insert at the end.)
            (NC.SetToLinks SourceCard (NCONC1 ToLinks Link)))
          (T (* Stick it at the front.)
            (NC.SetToLinks SourceCard (CONS Link ToLinks)
          [if (EQ (fetch (Link AnchorMode) of Link)
            'GlobalGlobal)
            then (NC.SetGlobalLinks SourceCard (CONS Link (NC.FetchGlobalLinks SourceCard)
              (NC.SetLinksDirtyFlg SourceCard T)
              Link])

```

**(NC.DelFromLink**

```

[LAMBDA (Link NoOrphanHookFlg) (* Randy.Gobbel "15-Jul-87 18:21")

  (** Delete a FromLink from its destination card Hook card to orphan if this is the last link.)

  (** rht 11/15/84%: Changed decision as to when to orphanize a card.
  Now must have deleted its last link (not just last subbox or filedcard link)%.
  Also checks that link doesn't point from card to itself.)

  (** rht 12/1/84%: Now doesn't do any work unless ID is valid, i.e.
  not DELETED or FREE.)

  (** kirk%: 13Nov85%: deleted of use of DatabaseStream and changed to call CachedDelFromLink.)

  (** kirk 29Jan86 Changed to a direct recursive call eliminating CachedDelFromLink.
  Added NC.FileInOrphanBox.)

  (** rht 2/14/86%: Now calls NC.UncacheLinks instead of two calls to NC.SetToLinks and NC.SetFromLinks.)

  (** kef 7/30/86%: Modified so that it only puts the FROMLINKS in the case of an inactiveP card.)

  (** rg [5/8/87] check for active card should have been check for links cached ;
  fixed)

  (** rg [7/15/87] now waits to call NC.PutFromLinks until after NC.FileInOrphanBox)

(LET ((DestinationCard (fetch (Link DestinationCard) of Link)))
 (COND
  ((NC.ValidCardP DestinationCard)
   (COND
    ((NC.LinksCachedP DestinationCard)
     (NC.CachedDelFromLink Link DestinationCard)
     (OR NoOrphanHookFlg (NC.FileInOrphanBox Link DestinationCard)))
    (T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of DestinationCard))
      (NC.GetLinks DestinationCard)
      (NC.CachedDelFromLink Link DestinationCard)
      (OR NoOrphanHookFlg (NC.FileInOrphanBox Link DestinationCard))
    )
  )

```

(NC.PutFromLinks DestinationCard)
(NC.UncacheLinks DestinationCard)
Link])

(NC.CachedDelFromLink

[LAMBDA (Link DestinationCard) (\* kirk%: "29-Jan-86 17:07")

(\* Delete a FromLink from its destination card Hook card to orphan if this is the last link.)
(\* rht 11/15/84%: Changed decision as to when to orphanize a card.
Now must have deleted its last link (not just last subbox or filedcard link)%.
Also checks that link doesn't point from card to itself.)
(\* rht 12/1/84%: Now doesn't do any work unless DestinationCard is valid, i.e.
not DELETED or FREE.)
(\* kirk%: 13Nov85%: copied out of NC.DelFromLink and changed to use NoteFile instead of DatabaseStream)
(\* kirk 14Nov85%: deleted use of LinkID)
(\* rht 12/9/85%: Changed Global var NC.TopLevelCards to call on the NC.FetchTopLevelCards fn.)
(\* kirk 29Jan86 moved all the orphans stuff to NC.FileInOrphanBox called by NC.DelFromLink)

(NC.SetFromLinks DestinationCard (DREMOVE (for OldLink in (NC.FetchFromLinks DestinationCard)
thereis (AND (NC.SameLinkP Link OldLink)
OldLink))
(NC.FetchFromLinks DestinationCard)))
(NC.SetLinksDirtyFlg DestinationCard T)
Link])

(NC.DelToLink

[LAMBDA (Link) (\* Randy.Gobbel "15-May-87 16:54")

(\* Delete ToLink spoeified by Link from cards on DatabasseStream)
(\* rht 12/1/84%: Now doesn't do any work unless SourceCard is valid, i.e.
not DELETED or FREE.)
(\* kirk%: 13Nov85%: changed to use NoteFile from card from link instead of DatabaseStream)
(\* rht 2/14/86%: Now calls NC.UncacheLinks instead of two calls to NC.SetToLinks and NC.SetFromLinks.)
(\* rg |5/15/87| now calls NC.LinksCachedP instead of NC.ActiveCardP)

(PROG ((SourceCard (fetch (Link SourceCard) of Link)))
(COND
((NOT (NC.ValidCardP SourceCard)))
((NC.LinksCachedP SourceCard)
(NC.SetToLinks SourceCard (DREMOVE (for OldLink in (NC.FetchToLinks SourceCard)
thereis (AND (NC.SameLinkP Link OldLink)
OldLink))
(NC.FetchToLinks SourceCard)))
(NC.SetLinksDirtyFlg SourceCard T))
(T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard)
"NC.AddFromLink")
(NC.GetLinks SourceCard)
(NC.SetToLinks SourceCard (DREMOVE (for OldLink in (NC.FetchToLinks SourceCard)
thereis (AND (NC.SameLinkP Link OldLink)
OldLink))
(NC.FetchToLinks SourceCard)))
(NC.SetLinksDirtyFlg SourceCard T)
(NC.PutLinks SourceCard))
(NC.UncacheLinks SourceCard)))
(RETURN Link]))

(NC.DelReferencesToCard

[LAMBDA (SourceCard LinkOrDestinationCard Don'tCreateDeletedImageObjFlg)
; Edited 3-Dec-87 19:00 by rht:

(\* The card specified by DestinationID is being deleted. Remove all references to it from the card specified by SourceCard)
(\* kirk%: 13Nov85%: deleted use of DatabaseStream)
(\* fgh |2/5/86| Added call to NC.ApplyFn)
(\* kef 7/31/86%: Added wrapping of write lock grabbing so that deactivating the card preserves the nesting of write lock
ownership.)
(\* kef 8/5/86%: Added putting of Main Card data in ActiveP case.
This is so that if subsequent other people bring up the card, they will see the fact that link card was deleted.)
(\* kef 8/7/86%: Now requires that the card is being edited rather than just Active to thwart direct write through to NoteFile.)

(\* rht |8/11/86| Added ShrunkenFlg so that SourceCard is reshunk afterwards if necessary.)

(\* fgh |8/30/86| Changed APPLY\* to NC.ApplyFn. Appears there might be trouble here since we don't appear to wait if we can't get all the write permission locks. I am assuming for now that this is being taken care of at a higher level.)

(\* rht 9/19/86%: It was testing NC.BeingEditedP to decide whether had to obtain write permission, etc. I think NC.ActiveCardP is enough. NC.BeingEditedP is bogus anyway as it doesn't work on card's whose editors don't have processes like graph cards.)

(\* rht 11/4/86%: Now takes Don'tCreateDeletedImageObjFlg arg.)

(\* rht 3/19/87%: Undo Ken's server-related fix of 8/5/86, i.e. ripped out call to NC.PutMainCardData when card is active.)

(\* rht 6/9/87%: Now marks card dirty after calling DeleteLinksFn.)

```
(LET ((ShrunkenFlg (NC.GetShrunkenWin SourceCard))
      NoteCardType Substance)
 [COND
  ((NC.ActiveCardP SourceCard)
   (SETQ NoteCardType (NC.RetrieveType SourceCard))
   (SETQ Substance (NC.FetchSubstance SourceCard))
   (NC.ApplyFn DeleteLinksFn SourceCard LinkOrDestinationCard Don'tCreateDeletedImageObjFlg)
   (NC.MarkCardDirty SourceCard))
  (T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
        (if (NC.ValidCardP (NC.GetNoteCard SourceCard))
            then (RESETLST
                  [RESETSAVE (for CardPart in '(SUBSTANCE TOLINKS GLOBALTOLINKS PROPLIST)
                              do (NC.ApplyFn ObtainWritePermissionFn SourceCard CardPart))
                   ` (NC.DeactivateCard ,SourceCard)
                   (SETQ NoteCardType (NC.RetrieveType SourceCard))
                   (SETQ Substance (NC.FetchSubstance SourceCard))
                   (NC.ApplyFn DeleteLinksFn SourceCard LinkOrDestinationCard
                               Don'tCreateDeletedImageObjFlg)
                   (NC.PutMainCardData SourceCard))))]
      (NC.DelReferencesToCardFromGlobalList SourceCard LinkOrDestinationCard)
      (AND ShrunkenFlg (SHRINKW (NC.FetchWindow SourceCard))
```

**(NC.DelReferencesToCardFromGlobalList**

[LAMBDA (SourceCard LinkOrDestinationCard) (\* rht%: " 5-Jun-87 14:56")

(\* LinkOrDestinationCard is being deleted. Remove all references to it from the global links list of SourceCard)

(\* rht 11/19/84%: Fixed a naming typo changing ID to SourceCard everywhere. Also fixed DestinationID arg to be LinkOrDestinationID.)

(\* kirk%: 14Nov85%: deleted use of of DatabaseStream and LinkID and changed EQs to NC.Same\*Ps)

(\* rg |5/15/87| now calls NC.LinksCachedP instead of NC.ActiveCardP)

(\* rht&pmi 6/5/87%: Added call to NC.UncacheLinks.)

```
(COND
  ((NC.LinksCachedP SourceCard)
   (NC.SetGlobalLinks SourceCard (for Link in (NC.FetchGlobalLinks SourceCard)
                                             when (if (NC.CardP LinkOrDestinationCard)
                                                       then (NOT (NC.SameCardP LinkOrDestinationCard
                                                                    (fetch (Link DestinationCard) of Link)))
                                                       else (NOT (NC.SameLinkP LinkOrDestinationCard Link)))
                                             collect Link)))
  (T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
        (NC.GetLinks SourceCard)
        (NC.SetGlobalLinks SourceCard (for Link in (NC.FetchGlobalLinks SourceCard)
                                                  when (if (NC.CardP LinkOrDestinationCard)
                                                            then (NOT (NC.SameCardP LinkOrDestinationCard
                                                                    (fetch (Link DestinationCard)
                                                                    of Link)))
                                                            else (NOT (NC.SameLinkP LinkOrDestinationCard Link)))
                                                  collect Link))
        (NC.PutLinks SourceCard)
        (NC.UncacheLinks SourceCard)))]
```

**(NC.DeleteLink**

[LAMBDA (Link NoOrphanHookFlg Don'tDelLinkIconFlg) ; Edited 9-Dec-88 18:53 by krivacic

;;; Delete a link with the option of not putting an orphan hook in case of last filing link. Also option of not deleting link icon from the source card's substance.

;;; fgh 5/2/86 Included calls to NC.DelReferencesToCardFromShowLinks in order to clean up ShowLinks windows when links are deleted.

;;; kef 7/17/86: Added the requirement that the write permissions be obtained for the TOLINKS of the Source Card and the FROMLINKS of the Destination Card.

;;; kef 7/22/86: Puts the links for the Destination Card now right away while we are still holding onto the write lock for the FROMLINKS.

;;; kef 7/30/86: Modified to check for Client's concept of whether he owns the write lock or not, thus deciding whether or not to setup the release of the write lock afterwards.



```

;;; fgh 8/30/86 Adpated to use NC.IfCardPartNotBusy. Note use of ERROR!. Unfortunately, it appears that the only way to keep TEDIT from deleting
;;; the Link's Image Object is to bail out but good. The control structure here works because NC.IfCardPartNotBusy returns NIL if the CardPart is busy.
;;; The Ts at the end as the last SExpr in each call to NC.IfCardPartNotBusy insure that NC.IfCardPartNotBusy returns non-NIL otherwise.

;;; rht 9/29/86: Removed the call to NC.PutFromLinks. Looks to me like it'll get called anyway by NC.DelFromLink if necessary.

;;; rht 10/6/86: Delete crossfilelink cards when their links are deleted.

;;; rht 11/4/86: Now takes Don'tCreateDeletedImageObjFlg arg.

;;; rht 11/13/86: Undid my change of 11/4/86.

;;; rg 3/11/87 changed call of NC.DeleteNoteCard to NC.DeleteNoteCardInternal

;;; rht 3/13/87: Broke out code into the NC.DeleteFromLink and NC.DeleteToLink functions.

;;; rht 5/26/87: Changed to match reduced functionality of NC.ValidLinkP, now checks that DestinationCard and SourceCard are valid cards.

;;; rht 6/9/87: Slight modification of previous fix; moved check for valid source and destination card to be around the relevant calls to NC.DeleteFromLink
;;; and NC.DeleteToLink.

;;; pmi 6/19/87: Leave the check for valid source and destination card for NC.DeleteFromLink and NC.DeleteToLink.

```

```

(AND (NC.ValidLinkP Link)
  (LET ((SourceCard (fetch (Link SourceCard) of Link))
        (DestinationCard (fetch (Link DestinationCard) of Link)))
    (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
      (OR (NC.IfCardPartNotBusy DestinationCard 'FROMLINKS (OR (NC.IfCardPartNotBusy
        DestinationCard
        'TOLINKS
        (NC.DeleteFromLink Link
          NoOrphanHookFlg)
        (NC.DeleteToLink Link
          Don'tDelLinkIconFlg)
        (replace (Link UID) of Link
          with -1)
        T)
        (ERROR!))
      T)
    (ERROR!)))]])

```

**(NC.DeleteToLink**

[LAMBDA (Link Don'tDelLinkIconFlg) (\* rht%: "27-May-87 14:40")

```

(* * This is part of what used to be the innards of NC.DeleteLink.)
(* * rht 3/26/87%: Changed to call NC.DeleteNoteCardInternal with QuietFlg on.)
(* * rht 5/27/87%: Changed to match reduced functionality of NC.ValidLinkP, now checks that SourceCard is valid card.
Now calls NC.DeleteCrossFileLinkCard which tries to delete other end of cross-file link if that notefile is open.)

```

```

(AND (NC.ValidLinkP Link)
  (LET ((SourceCard (fetch (Link SourceCard) of Link)))
    (AND (NC.ValidCardP SourceCard)
      (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
        (OR Don'tDelLinkIconFlg (NC.DelReferencesToCard SourceCard Link))
        (NC.DelToLink Link)
        (NC.DelReferencesToCardFromShowLinks SourceCard Link)
      )
    )
    (* * Delete cross file link cards when their links are deleted.)

    (if (NC.CrossFileLinkCardP SourceCard)
      then (NC.DeleteCrossFileLinkCard SourceCard)))]])

```

**(NC.DeleteFromLink**

[LAMBDA (Link NoOrphanHookFlg) (\* rht%: "27-May-87 14:40")

```

(* * This is part of what used to be the innards of NC.DeleteLink.)
(* * rht 3/26/87%: Changed to call NC.DeleteNoteCardInternal with QuietFlg on.)
(* * rht 5/26/87%: Changed to match reduced functionality of NC.ValidLinkP, now checks that DestinationCard is a valid
card. Now calls NC.DeleteCrossFileLinkCard which tries to delete other end of cross-file link if that notefile is open.)

```

```

(AND (NC.ValidLinkP Link)
  (LET ((DestinationCard (fetch (Link DestinationCard) of Link)))
    (AND (NC.ValidCardP DestinationCard)
      (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of DestinationCard))
        (NC.DelFromLink Link NoOrphanHookFlg)
        (NC.DelReferencesToCardFromShowLinks DestinationCard Link)
      )
    )
    (* * Delete cross file link cards when their links are deleted.)

    (if (NC.CrossFileLinkCardP DestinationCard)

```

then (NC.DeleteCrossFileLinkCard DestinationCard)))]

**(NC.DeletedLinkDisplayFn**

```
[LAMBDA (ImageObj ImageStream ImageStreamType HostStream) ; Edited 18-Aug-88 16:19 by pmi
;; pmi 8/18/88: Completely rewrote so that Deleted icons would hardcopy. They were being skipped entirely.
(DECLARE (GLOBALVARS NC.LinkIconSpaceInnerX NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerY
NC.LinkIconSpaceOuterY NC.LinkIconFont))
(LET ((Scale (DSPSCALE NIL ImageStream))
(Font NC.LinkIconFont)
ImageBox FontDescent ScaledSpaceOuterX ScaledSpaceOuterY Bottom Left BoxWidth BoxHeight Bitmap
BitmapDSP)
;; TEdit will call this function with ImageStreamType set. Here we take advantage of the fact that TEdit always recalculates the BOUNDBOX
;; of an imageobj, so it is safe to use the value stored as the BOUNDBOX property. Otherwise, we must calculate it ourselves to be on the
;; safe side.
(SETQ ImageBox (if ImageStreamType
then (OR (IMAGEOBJPROP ImageObj 'BOUNDBOX)
(NC.DeletedLinkImageBoxFn ImageObj ImageStream))
else (NC.DeletedLinkImageBoxFn ImageObj ImageStream)))
(SETQ FontDescent (FONTDESCENT Font))
(SETQ ScaledSpaceOuterX (TIMES Scale NC.LinkIconSpaceOuterX))
(SETQ ScaledSpaceOuterY (TIMES Scale NC.LinkIconSpaceOuterY))
(SETQ Bottom (PLUS (DIFFERENCE (DSPYPOSITION NIL ImageStream)
(fetch (IMAGEBOX YDESC) of ImageBox))
ScaledSpaceOuterY))
(SETQ Left (PLUS (DSPXPOSITION NIL ImageStream)
ScaledSpaceOuterX))
;; BoxHeight and BoxWidth are used to make a bitmap of the deleted icon, and therefore shouldn't be scaled.
(SETQ BoxHeight (IPLUS NC.LinkIconSpaceInnerY NC.LinkIconSpaceInnerY (FONTASCENT Font)
FontDescent))
(SETQ BoxWidth (IPLUS NC.LinkIconSpaceInnerX NC.LinkIconSpaceInnerX (STRINGWIDTH "Deleted" Font)))
(SETQ Bitmap (BITMAPCREATE BoxWidth BoxHeight))
(SETQ BitmapDSP (DSPCREATE Bitmap))
;; Move to the appropriate place in the bitmap to print the text.
(DSPXOFFSET NC.LinkIconSpaceInnerX BitmapDSP)
(DSPYOFFSET (PLUS NC.LinkIconSpaceInnerY FontDescent)
BitmapDSP)
(DSPFONT Font BitmapDSP)
(PRIN1 "Deleted" BitmapDSP)
;; Shade the bitmap so that the text is white and the background is black.
[BLTSHADE BLACKSHADE Bitmap NIL NIL NIL NIL (COND
(EQ (DSOPERATION NIL ImageStream)
'ERASE)
'ERASE)
(T 'INVERT)]
;; Send the deleted icon, in the form of a bitmap, to the image stream.
(BITBLT Bitmap 0 0 ImageStream Left Bottom)])
```

**(NC.DeletedLinkGetFn**

```
[LAMBDA (InputStream TextStream) (* rht%: "10-Jun-87 14:56")
(* rht 6/10/87%: Now skips carriage return if any.)
(if (EQ (\PEEKBIN InputStream)
13)
then (BIN InputStream)
(BIN InputStream)
NC.DeletedLinkImageObject])
```

**(NC.DeletedLinkImageBoxFn**

```
[LAMBDA (ImageObj ImageStream CurrentX RightMargin) ; Edited 11-Oct-88 18:09 by Randy.Gobbel
;; pmi 8/18/88: Completely rewrote so that Deleted icons would hardcopy. They were being skipped entirely.
(DECLARE (GLOBALVARS NC.LinkIconSpaceInnerX NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerY
NC.LinkIconSpaceOuterY NC.LinkIconFont))
(LET ((Scale (DSPSCALE NIL ImageStream))
(Font NC.LinkIconFont)
(create IMAGEBOX
XSIZE _ (TIMES Scale (PLUS NC.LinkIconSpaceInnerX NC.LinkIconSpaceInnerX NC.LinkIconSpaceOuterX
NC.LinkIconSpaceOuterX (STRINGWIDTH "Deleted" Font)))
YSIZE _ (TIMES Scale (PLUS NC.LinkIconSpaceInnerY NC.LinkIconSpaceInnerY NC.LinkIconSpaceOuterY
NC.LinkIconSpaceOuterY (FONTASCENT Font)
(FONTDESCENT Font)))
YDESC _ (COND
(RightMargin
;; This is in a TEdittextstream .
(TIMES Scale (PLUS (FONTDESCENT Font)
NC.LinkIconSpaceInnerY NC.LinkIconSpaceOuterY)))
```

```
(T 0))
XKERN _ 0])
```

**(NC.DeletedLinkPutFn**

```
[LAMBDA (ImageObj OutputStream) (* fgh%: "26-Feb-84 01:04")
(BOUT OutputStream 1])
```

**(NC.DeletedLinkImageObjP**

```
[LAMBDA (ImageObj) (* rht%: "12-Jun-85 10:07")
```

(\* Returns non-nil if ImageObj is a deleted link icon.)

```
(AND (IMAGEOBJP ImageObj)
(EQ (FUNCTION NC.DeletedLinkDisplayFn)
(IMAGEOBJPROP ImageObj 'DISPLAYFN]))
```

**(NC.DeleteBadLink**

```
[LAMBDA (Link ImageObj ImageStream StreamType TextStream Scale)
; Edited 30-May-88 12:07 by Trigg
```

;; pmi&rht 6/19/87: Deletes a bad link when it is discovered at display time. If the link icon being displayed is in the Show Links window, then find its corresponding real link and delete it. Otherwise, just delete the link.

;; rht 5/30/88: Now checks value of NC.UseDeletedLinkIconIndicatorsFlg. If nil, then marks the link for deletion.

```
(DECLARE (GLOBALVARS NC.UseDeletedLinkIconIndicatorsFlg))
(LET ((Card (fetch (Link SourceCard) of Link)))
(COND
((LISTGET (fetch (Link UserData) of Link)
'InsidePropListEditor)
(for RealLink in (NC.FetchFromLinks Card) when (NC.SameLinkP RealLink Link)
do (NC.DeleteLink RealLink))
(for RealLink in (NC.FetchToLinks Card) when (NC.SameLinkP RealLink Link) do (NC.DeleteLink RealLink))
'DON'T)
(NC.UseDeletedLinkIconIndicatorsFlg ; Change imageobj into deleted link icon.
(NC.DeleteLink Link NIL T)
(NC.ReplaceWithDeletedLinkImageObj ImageObj)
(APPLY* (IMAGEOBJPROP ImageObj 'DISPLAYFN)
ImageObj ImageStream StreamType TextStream Scale)
'CHANGED)
(T ; Mark the link as needing deletion. NC.EditNoteCard will remove it once we return from the card type's editfn.
(NC.SetUserDataProp Card 'BadLinks (CONS Link (NC.FetchUserDataProp Card 'BadLinks))
```

**(NC.ReplaceWithDeletedLinkImageObj**

```
[LAMBDA (ImageObject) (* pmi%: "19-Jun-87 10:28")
```

(\* pmi 6/19/87%: smash a bad link's image object with the fields of a "Deleted" image object.)

```
(DECLARE (GLOBALVARS NC.DeletedLinkImageObject))
(create IMAGEOBJ smashing ImageObject OBJECTDATUM _ (fetch (IMAGEOBJ OBJECTDATUM) of NC.DeletedLinkImageObject)
IMAGEOBJPLIST _ (fetch (IMAGEOBJ IMAGEOBJPLIST) of NC.DeletedLinkImageObject)
IMAGEOBJFNS _ (fetch (IMAGEOBJ IMAGEOBJFNS) of NC.DeletedLinkImageObject))
```

**(NC.CheckForOrphanDelete**

```
[LAMBDA (SourceCard DestinationCard SourceWindow) (* rht%: "9-Dec-87 16:52")
```

(\* Check to make sure we are not deleting the last reference to a note card from the orphan card. Last references are okay to delete it since they will be put in the orphan card. But deleting last reference from the orphan card is very, very bad. Warn the user.)

(\* kirk%: 14Nov85%: changed EQs to NC.Same\*Ps)

(\* rht 7/4/86%: Brought up to date with 1.3k. E.g. took out mention of NC.OrphansID)

(\* rg [5/15/87] now calls NC.LinksCachedP instead of NC.ActiveCardP)

(\* rht 12/9/87%: No longer checks links caching. Now just assumed that links are cached when this fn is called.)

```
(DECLARE (GLOBALVARS NC.SubBoxLinkLabel NC.FiledCardLinkLabel))
(LET [(FilingLinks (LIST NC.SubBoxLinkLabel NC.FiledCardLinkLabel))
(OrphansCard (fetch (NoteFile OrphansCard) of (fetch (Card NoteFile) of SourceCard))
(AND (NC.SameCardP SourceCard OrphansCard)
(for Link in (NC.FetchFromLinks DestinationCard) when (FMEMB (fetch (Link Label) of Link)
FilingLinks)
unless (NC.SameCardP (fetch (Link SourceCard) of Link)
OrphansCard)
never Link)
(NC.PrintMsg (NC.FetchWindow SourceCard)
T "You have just deleted the last filing link to " (NC.RetrieveTitle DestinationCard)
(CHARACTER 13)
"The Search operation can be used to find it."))
```

**(NC.FileInOrphanBox**

[LAMBDA (Link DestinationCard)

(\* rht%: "29-May-87 01:00")

(\* \* Hook card to orphan if this is the last link.)

(\* \* rht 7/4/86%: Changed to call NC.FetchSpecialCards so that RegistryCard, etc. won't have to be filed somewhere.)

(\* \* rg |11/18/86| Added globalvars declaration)

(\* \* rht 4/20/87%: Changed to work on new cards unless they're displayed on screen. Also removed reference to long-demised NC.OrphansID.)

(\* \* rht 5/28/87%: Now calls NC.CardNeedsFilingP, i.e. card should need filing before we gripe.)

```
(LET ((SourceCard (fetch (Link SourceCard) of Link)))
[AND (NC.CardNeedsFilingP DestinationCard)
(NOT (NC.SameCardP SourceCard DestinationCard))
(NOT (AND (NC.FetchNewCardFlg DestinationCard)
(NC.FetchWindow DestinationCard)))
[NOT (NC.SameCardP SourceCard (fetch (NoteFile OrphansCard) of (fetch (Card NoteFile) of SourceCard)
(COND
((OR (NULL (NC.FetchFromLinks DestinationCard))
(for FromLink in (NC.FetchFromLinks DestinationCard)
always (NC.SameCardP (fetch (Link SourceCard) of FromLink)
DestinationCard)))
(NC.PrintMsg NIL T "You have just removed the last link to " (NC.RetrieveTitle DestinationCard)
)
" ."
(CHARACTER 13)
"It is being filed in the Orphan FileBox.")
(NC.HookToOrphanCard DestinationCard)
([for FromLink in (NC.FetchFromLinks DestinationCard) never (FMEMB (fetch (Link Label)
of FromLink)
'(SubBox FiledCard]
(NC.PrintMsg NIL T "You have just unfiled " (NC.RetrieveTitle DestinationCard)
" from its last filebox."
(CHARACTER 13)
"It is being filed in the Orphan FileBox.")
(NC.HookToOrphanCard DestinationCard]
Link])
```

**(NC.HookToOrphanCard**

[LAMBDA (Card OrphansCard)

(\* Randy.Gobbel "19-Feb-87 16:57")

(\* The last reference to the card has just been deleted. Hook this card to the orphan card so it doesn't get lost forever)

(\* \* kirk%: 14Nov85%: deleted use of DatabaseStream and changed EQs to NC.Same\*Ps)

(\* \* kef 7/29/86%: Took out the calls to NC.PutMainCardData, NC.PutLinks, and NC.DeactivateCard after NC.MakeFilingLink. The reason is that this is done within NC.InsertLinkBeforeMarker or NC.InsertLinkInOrdering, which are called by NC.MakeFilingLink. Therefore, don't do this twice.)

(\* \* rg |2/19/87| Return parent as value)

```
(DECLARE (GLOBALVARS NC.FiledCardLinkLabel NC.SubBoxLinkLabel))
(LET ((NoteFile (fetch (Card NoteFile) of Card)))
(OR OrphansCard (SETQ OrphansCard (fetch (NoteFile OrphansCard) of NoteFile)))
(COND
((NC.ActiveCardP OrphansCard)
(COND
((NC.MakeFilingLink OrphansCard Card (if (NC.FileBoxP Card T)
then NC.SubBoxLinkLabel
else NC.FiledCardLinkLabel))
OrphansCard)))
(T (WITH.MONITOR (NC.FetchMonitor NoteFile)
(NC.GetNoteCard OrphansCard)

(* These no longer needed expressions used to go after the NC.MakeFilingLink below%:
(NC.PutMainCardData OrphansCard) (NC.PutLinks OrphansCard)
(NC.DeactivateCard OrphansCard)

(COND
((NC.MakeFilingLink OrphansCard Card (if (NC.FileBoxP Card T)
then NC.SubBoxLinkLabel
else NC.FiledCardLinkLabel))
OrphansCard))))))
```

**(NC.InsertLinkBeforeMarker**

[LAMBDA (SourceCard DestinationCard LinkLabel DisplayMode Marker# NoSpacerFlg)

; Edited 3-Dec-87 19:00 by rht:

(\* Insert a link to DestinationCard in SourceCard just before the

Marker#'th place marker object)

- (\* rht 12/7/84%: Now returns the newly created link.)
- (\* rht 9/13/85%: When card is inactive, there won't be a current cursor position. So if no marker# is specified, then insert at end.)
- (\* kirk%: 14Nov85%: deleted use of DatabaseStream)
- (\* kef 7/28/86%: Added call to obtain write locks on SUBSTANCE and all LINKS in the case when the card is not NC.ActiveCardP.)
- (\* kef 7/31/86%: Added "balancing" of write lock ownership with NC.DeactivateCard.)
- (\* kef 8/7/86%: Now cached changes only occur if the card is currently being edited.)
- (\* fgh |8/30/86| Adpated to NC.IfCardPartsNotBusy)
- (\* fgh |9/1/86| Rescinded kef's change of |8/7/86.| This causes active but nit visible cards to be written over even if they are dirty.)

```
(LET (Objects TextStream (Spacer (CONCAT (CHARACTER 13)))
      Link BusyPart)
[COND
  [(NC.ActiveCardP SourceCard)
   (NC.IfMultipleCardPartsNotBusy SourceCard (SUBSTANCE TOLINKS FROMLINKS GLOBALTOLINKS)
    (SETQ TextStream (NC.FetchSubstance SourceCard))
    [COND
      [[AND (FIXP Marker#)
             (SETQ Objects (NC.TEditCollectObjects TextStream (FUNCTION NC.PlaceMarkerP)
              (COND
                ((EQ Marker# 0)
                 (TEDIT.SETSEL TextStream 1 0 'LEFT))
                ((IGREATERP Marker# (FLENGTH Objects))
                 (TEDIT.SETSEL TextStream (ADD1 (TEDIT.NCHARS TextStream))
                  0
                  'RIGHT))
                (T (TEDIT.SETSEL TextStream (CADAR (FNTH Objects Marker#))
                  0
                  'LEFT]
              (NC.MarkersInFileBoxesFlg (TEDIT.SETSEL TextStream (GETEOFPTR TextStream)
                0
                'RIGHT]
            (SETQ Link (NC.InsertLinkInText TextStream LinkLabel DestinationCard SourceCard DisplayMode))
            (COND
              ((NULL NoSpacerFlg)
               (TEDIT.INSERT TextStream Spacer]
            (T (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
                [NC.IfMultipleCardPartsNotBusy SourceCard (SUBSTANCE TOLINKS FROMLINKS GLOBALTOLINKS)
                 (COND
                   ((NC.CardP (NC.GetNoteCard SourceCard))
```

(\* This next set of obtain write locks is only to make sure that we come out even when NC.DeactivateCard releases the write locks. Note that the SourceCard will be deactivated upon exit of the RESETLST, as set up by the RESETSAVE.)

```
[RESETSAVE (for CardPart in '(SUBSTANCE TOLINKS GLOBALTOLINKS PROPLIST)
            do (APPLY* (fetch (Card ObtainWritePermissionFn) of SourceCard)
                      SourceCard CardPart))
  (NC.DeactivateCard ,SourceCard]
(SETQ TextStream (NC.FetchSubstance SourceCard))
[COND
  [[AND (FIXP Marker#)
         (SETQ Objects (NC.TEditCollectObjects TextStream (FUNCTION
                                                             NC.PlaceMarkerP]
        (COND
          ((EQ Marker# 0)
           (TEDIT.SETSEL TextStream 1 0 'LEFT))
          ((IGREATERP Marker# (FLENGTH Objects))
           (TEDIT.SETSEL TextStream (ADD1 (TEDIT.NCHARS TextStream))
            0
            'RIGHT))
          (T (TEDIT.SETSEL TextStream (CADAR (FNTH Objects Marker#))
            0
            'LEFT]
        (T
          (* Cursor has no position in closed Tedit win, so just set
            selection to end.)
          (TEDIT.SETSEL TextStream (ADD1 (TEDIT.NCHARS TextStream))
            0
            'RIGHT]
        (SETQ Link (NC.InsertLinkInText TextStream LinkLabel DestinationCard SourceCard
          DisplayMode))
        (COND
          ((NULL NoSpacerFlg)
           (TEDIT.INSERT TextStream Spacer)))
        (NC.PutMainCardData SourceCard)
        (NC.PutLinks SourceCard)])
```

Link])

**(NC.LinkLabelP**

```
[LAMBDA (Link ListOfLabels) (* kirk%: "15-Nov-85 01:43")
  (* * rht 9/4/84%: Modified so as to return the linklabel if found or nil.)
  (CAR (FMEMB (fetch (Link Label) of Link)
    (OR (LISTP ListOfLabels)
      (LIST ListOfLabels]))
```

**(NC.IDAlphOrder**

```
[LAMBDA (LinkOrCard1 LinkOrCard2) (* rht%: "18-Jul-86 15:20")
  (* * Return T if Card1's title is alphabetically before Card2's)
  (* * rht 7/18/86%: Now can accept either links or cards as arg.)
  (ALPHORDER (NCP.CardTitle (if (NC.ValidLinkP LinkOrCard1)
    then (fetch (Link DestinationCard) of LinkOrCard1)
    else LinkOrCard1))
    (NCP.CardTitle (if (NC.ValidLinkP LinkOrCard2)
    then (fetch (Link DestinationCard) of LinkOrCard2)
    else LinkOrCard2]))
```

**(NC.MakeLink**

```
[LAMBDA (Window LinkLabel DestinationCard SourceCard DisplayMode AnchorMode Message NoDisplayFlg
  LinkToInsertAfter CrossFileLinksMode) ; Edited 24-Feb-88 15:03 by pmi
```

```
;;; Make a link from (OR Window SourceCard) to DestinationCard with linklabel of LinkLabel
;; rht 1/12/85: If need to create a new card, then now shows card type menu near window of SourceID.
;; rht 1/13/85: Added extra args Message and NoDisplayFlg.
;; rht 3/26/85: Added LinkToInsertAfter arg which should be NIL or a link to insert the new To link after. If NIL, then insert at front of ToLinks.
;; kirk 9/23/85: took out GETPROMPTWINDOW call for asknotecardtype
;; kirk: 14Nov85: changed NC.CoerceToID to to NC.CoerceToCard
;; fgh 11/16/85 Changed from PROG to LET and used COND to contyrol returned value.
;; fgh 2/5/86 Changed call DefaultLinkDisplayMode to FetchLinkDisplayMode
;; fgh 6/5/86 Now calls AskLinkLabel if LinkLabel arg is NIL
;; rht 7/4/86: Added check for readonly card.
;; kef 7/17/86: Added calls to grab the write permission on the appropriate card parts.
;; kef 7/22/86: Saves the links on the Destination Card now right away, while still holding onto the FROMLINKS write lock.
;; fgh 8/30/86 Adpated to use NC.IfCardPartNotBusy.
;; rht 9/29/86: Tossed Ken's call to NC.PutFromLinks; It was the cause of too many nasty breaks. Also made syntactic fixes.
;; rht 10/4/86: Now handles cross file links. New arg CrossFileLinksMode determines whether cross-file link will be two-way, i.e. will destination
;; card know it's being linked to.
;; rht 11/10/86: Now creates new crossfile link if Destination card is a CrossFileLink card that we didn't just create.
;; rht 11/14/86: Now checks if non-nil DestinationCard before trying to do cross-filelink stuff.
;; pmi 12/5/86: Modified message to NC.SelectNoteCards to mention SHIFT-selection.
;; rht 12/9/86: Throws out JustCreatedFlg marker stuff.
;; pmi 12/12/86: Removed obsolete ReturnLinksFlg argument in call to NC.SelectNoteCards.
;; rht 12/16/86: Fixed bug whereby electing not to open notefile containing crossfilelink dest card caused break.
;; rht 12/16/86: Now passes Window down to NC.GetCrossFileLinkDestCard.
;; rg 3/18/87 added NCP.WithLockedCards wrapper
;; rht 5/25/87: No longer tries to make two way cross-file links when dest notefile is open read-only. Also assumes that CrossFileLinksMode is one
;; of TWOWAY, ONEWAY or ASK. Now fills in the new RemoteCrossFileLinkCardUID field of cross file link cards.
;; rht 11/20/87: Now recomputes ShowLinks menus if they're visible.
;; rht 2/22/88: Now passes NIL argument to NC.AskNoteCardType which makes type selection menu come up at cursorspos.
;; pmi 2/24/88: Now passes Window as InterestedWindow argument to NC.MakeNoteCard.
(DECLARE (GLOBALVARS NC.SelectingSingleCardMenu NC.NewCrossFileLinksMode))
(OR SourceCard (SETQ SourceCard (NC.CoerceToCard Window)))
(AND
  (NC.CheckForNotReadOnly SourceCard Window "Can't make links in ")
  (NCP.WithLockedCards
    (LET
      (Link Type)
      (OR Window (SETQ Window (NC.FetchWindow SourceCard)))
      (OR Message (SETQ Message "Please shift-select the Card or Box to be linked to.))
      (OR LinkLabel (SETQ LinkLabel (NC.AskLinkLabel Window NIL NIL T NIL)))
      (OR DestinationCard (SETQ DestinationCard (NC.SelectNoteCards T [FUNCTION (LAMBDA (Card)
        (COND
          ((NOT (NC.SameCardP Card
            SourceCard))
            T)
          (T (NC.PrintMsg
```

```

" (CHARACTER 13)
Window T "A Card/Box
cannot link to itself.
"Selection ignored."
(CHARACTER 13))
NIL]
NC.SelectingSingleCardMenu SourceCard Message)))

```

```

[if (EQ DestinationCard '*New% Card*)
  then (SETQ DestinationCard (AND (SETQ Type (NC.AskNoteCardType))
    (NC.CoerceToCard (NC.MakeNoteCard Type (fetch (Card NoteFile)
      of SourceCard)
      NIL NoDisplayFlg NIL NIL Window]

```

;;; If we're trying to link to a CrossFileLink card, then check whether card was just created. If so, then it's the first link, otherwise we make a new CrossFileLink.

```

(AND DestinationCard (NC.CrossFileLinkCardP DestinationCard)
  (SETQ DestinationCard (NC.GetCrossFileLinkDestCard DestinationCard Window))
  (NC.SetUserDataProp DestinationCard 'JustCreatedFlg NIL))
(if DestinationCard
  then (NC.IfCardPartNotBusy
    DestinationCard
    'FROMLINKS
    (NC.IfCardPartNotBusy
      SourceCard
      'TOLINKS

```

;;; If have cross-file link, then make two new crossfilelink cards, one per notefile. Make global link over there from crossfilelink card to DestinationCard and local link here from SourceCard to crossfilelink card.

```

[if (NOT (NC.SameNoteFileP (fetch (Card NoteFile) of SourceCard)
  (fetch (Card NoteFile) of DestinationCard)))
  then (LET ([CrossFileLinksTwoWayFlg (OR (EQ CrossFileLinksMode 'TWOWAY)
    (AND (NULL CrossFileLinksMode)
      (EQ NC.NewCrossFileLinksMode 'TWOWAY))
    (AND [OR (EQ CrossFileLinksMode 'ASK)
      (AND (NULL CrossFileLinksMode)
        (EQ NC.NewCrossFileLinksMode
          'ASK]
      (NC.AskCrossFileLinkMode DestinationCard
        Window]
    RemoteSourceCard)
  (if CrossFileLinksTwoWayFlg
    then (AND (SETQ RemoteSourceCard (NC.CreateCrossFileLinkCard DestinationCard
      SourceCard T))
      (NC.MakeGlobalLink Window LinkLabel DestinationCard RemoteSourceCard
        DisplayMode)))
  (SETQ DestinationCard (NC.CreateCrossFileLinkCard SourceCard DestinationCard
    CrossFileLinksTwoWayFlg))
  (if RemoteSourceCard
    then ; Make the two crossfile link cards know about each other's
      ; UIDs.
      (replace (CrossFileLinkSubstance RemoteCrossFileLinkCardUID)
        of (NC.FetchSubstance RemoteSourceCard) with (fetch (Card UID)
          of DestinationCard))
      (replace (CrossFileLinkSubstance RemoteCrossFileLinkCardUID)
        of (NC.FetchSubstance DestinationCard) with (fetch (Card UID)
          of RemoteSourceCard]
  [SETQ Link (create Link
    UID _ (NC.MakeUID)
    SourceCard _ SourceCard
    DestinationCard _ DestinationCard
    AnchorMode _ AnchorMode
    Label _ LinkLabel
    DisplayMode _ (OR DisplayMode (NC.FetchLinkDisplayMode SourceCard]
  (NC.AddToLink Link LinkToInsertAfter)
  (NC.AddFromLink Link) ; Recompute ShowLinks menus if they're up.
  (if (NC.FetchShowLinksWindow SourceCard)
    then (NC.ShowLinks SourceCard))
  (if (NC.FetchShowLinksWindow DestinationCard)
    then (NC.ShowLinks DestinationCard))
  Link))
else NIL])

```

### (NC.CachedMakeFilingLink

```

[LAMBDA (SourceCard DestinationCard LinkType DisplayMode IgnoreMarkersFlg)
  (* rht%:"13-Oct-86 12:31")

```

(\* Assumes SourceCard is cached. Inserts a link to DestinationCard into SourceCard. Unless IgnoreMarkersFlg is non-nil or there's no marker for type of DestinationCard, then will insert somewhere between appropriate marker and either the next marker or the end of SourceCard substance using SourceCard's OrderingFn to determine where to insert. If there's no OrderingFn, but there is an appropriate marker, then stick at end of group. If no marker and no OrderingFn, then stick wherever cursor is.)

```

(LET ((OrderingFn (NC.GetProp SourceCard 'OrderingFn))
  (TextStream (NC.FetchSubstance SourceCard))

```

```

StartLoc EndLoc FoundMarkerFlg Link)
[OR IgnoreMarkersFlg (SETQ FoundMarkerFlg (for MarkerPairs on (NC.TeditCollectObjects
TextStream
(FUNCTION NC.PlaceMarkerP))
eachtime (BLOCK)
do (if (NC.MarkerMatchesCardP (CAAR MarkerPairs)
DestinationCard)
then (SETQ StartLoc (CADAR MarkerPairs))
[SETQ EndLoc (OR (CADADR MarkerPairs)
(ADD1 (GETEOFPTR TextStream)
(RETURN T))
finally (SETQ StartLoc 1)
(SETQ EndLoc (ADD1 (GETEOFPTR TextStream)
[COND
[OrderingFn (for ObjectPair in (NC.TeditCollectObjects TextStream (FUNCTION NC.LinkIconImageObjP))
eachtime (BLOCK) until (GREATERP (CADR ObjectPair)
EndLoc)
when [AND (LEQ StartLoc (CADR ObjectPair))
(APPLY* OrderingFn DestinationCard (fetch (Link DestinationCard)
of (NC.FetchLinkFromLinkIcon
(CAR ObjectPair))
do (TEDIT.SETSEL TextStream (CADR ObjectPair)
0
'LEFT)
(RETURN)
finally (TEDIT.SETSEL TextStream EndLoc 0 'RIGHT]
(FoundMarkerFlg (TEDIT.SETSEL TextStream EndLoc 0 'RIGHT]
(PROG1 (NC.InsertLinkInText TextStream LinkType DestinationCard SourceCard DisplayMode)
(TEDIT.INSERT TextStream NC.FileBoxLinkSpacer])

```

**(NC.MakeFilingLink**

[LAMBDA (SourceCard DestinationCard LinkType) ; Edited 3-Dec-87 19:00 by rht:

(\* Make a link from FileBox to DestinationCard If card is a regular card, insert link at end of contents card. If card is a supercontents of contents card then insert just before the subcontents marker (i.e., at end of super contents list)%. If card is a subcontents card then insert just before note cards marker (i.e., at end of subcontents list))

(\* rht 10/25/85%: Now checks if card's window is shrunken. If so, shrink it back at the end.)

(\* kirk%: 14Nov85%: deleted use of DatabaseStream)

(\* fgh |11/15/85| Adapted from and intended to replace NC.MakeAContentsHook)

(\* kirk 11Apr86 took out hard coded link display mode)

(\* rht 10/13/86%: Totally overhauled. Now calls NC.CachedMakeFilingLink which does both inserting before markers and inserting in ordering.)

```

(if (NC.ActiveCardP SourceCard)
then [NC.IfMultipleCardPartsNotBusy SourceCard (SUBSTANCE TOLINKS FROMLINKS GLOBALTOLINKS)
(LET ((ShrunkenFlg (NC.GetShrunkenWin SourceCard)))
(PROG1 (NC.CachedMakeFilingLink SourceCard DestinationCard LinkType)
(AND ShrunkenFlg (SHRINKW (NC.FetchWindow SourceCard)))))]
else (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of SourceCard))
[NC.IfMultipleCardPartsNotBusy SourceCard (SUBSTANCE TOLINKS FROMLINKS GLOBALTOLINKS)
(if (NC.CardP (NC.GetNoteCard SourceCard))
then
(* * This next set of obtain write locks is only to make sure that we come out even when NC.DeactivateCard releases the
write locks. Note that the SourceCard will be deactivated upon exit of the RESETLST, as set up by the RESETSAVE.)
[RESETSAVE (for CardPart in '(SUBSTANCE TOLINKS GLOBALTOLINKS PROPLIST)
do (APPLY* (fetch (Card ObtainWritePermissionFn) of SourceCard)
SourceCard CardPart)
'(NC.DeactivateCard , SourceCard]
(PROG1 (NC.CachedMakeFilingLink SourceCard DestinationCard LinkType)
(NC.PutMainCardData SourceCard)
(NC.PutLinks SourceCard)))]])

```

**(NC.MakeFilingLinks**

[LAMBDA (Card Msg InterestedWindow) (\* Randy.Gobbel " 2-Apr-87 15:38")

(\* Hooks card specified by Card to all of the current contents cards by a Contents link)

(\* rht 8/1/84%: Changed the NC.PrintMsg2 call for "No FileBox has been specified." to use NIL as second arg rather than T. This prevents erasure of previous error messages.)

(\* rht 12/8/84%: Massive rewrite. Now calls NC.MakeChildLink. And always orphanizes if no parent specified. This is because it's currently called only by NC.InsureProperFiling.)

(\* rht 6/25/85%: Now returns NewParents and checks if CANCELLED comes back from NC.SelectNoteCards so can abort the calling operation.)

(\* Fgh |11/15/85| Adapted from and intended to replace NC.MakeContentsHooks)



- (\* \* fgh |8/6/86| Updated to use NC.AttachPromptWindow and added Msg arg)
- (\* \* pmi 12/5/86%: Modified message to NC.SelectNoteCards to mention SHIFT-selection.)
- (\* \* pmi 12/12/86%: Removed obsolete ReturnLinksFlg argument in call to NC.SelectNoteCards.)
- (\* \* rg |2/19/87| Make sure NewParents always has the real value)
- (\* \* rg |3/18/87| added NC.CardSelectionOperation wrapper)
- (\* \* rht 3/23/87%: Changed so that message will fit on one less line.)
- (\* \* rht 3/23/87%: Now takes InterestedWindow arg.)
- (\* \* Rht 3/24/87%: Now calls NC.CoerceToInterestedWindow)
- (\* \* rht 3/26/87%: No longer allows selection of boxes in foreign notefiles.)
- (\* \* rg |4/1/87| changed CANCELLED to DON'T)

```
(DECLARE (GLOBALVARS NC.SelectingContentsMenu))
(NCP.WithLockedCards (PROG ((NoteFile (fetch (Card NoteFile) of Card))
    OneHook NewParents)
  (OR InterestedWindow (SETQ InterestedWindow (NC.CoerceToInterestedWindow Card)))
  (SETQ NewParents (NC.SelectNoteCards NIL [FUNCTION (LAMBDA (SelectedCard)
    (AND
      (NC.FileBoxP SelectedCard)
      (NC.SameNoteFileP
        NoteFile
        (fetch (Card NoteFile)
          of SelectedCard]
        NC.SelectingContentsMenu Card (CONCAT (OR Msg "")
          ("Done' with no
          selections files in
          ToBeFiled.")")
        T)))
    [COND
      ([NOT (OR (EQ NewParents 'DON'T)
        (AND NewParents (for ParentCard in NewParents bind OneHook
          when (NC.MakeChildLink Card ParentCard
            InterestedWindow)
          do (SETQ OneHook T) finally (RETURN OneHook]
        (NC.PrintMsg InterestedWindow NIL "No FileBox has been specified."
          (CHARACTER 13)
          "This card will be filed in the ToBeFiled Box."
          (CHARACTER 13))
        (SETQ NewParents (NC.HookToOrphanCard Card (fetch (NoteFile ToBeFiledCard)
          of NoteFile)))
        (COND
          ((NC.AttachPromptWindow InterestedWindow NIL NIL NIL T)
            (SPAWN.MOUSE)
            (DISMISS 2000)
            (NC.ClearMsg InterestedWindow T]
        (RETURN NewParents])
```

**(NC.RelabelLink**

```
[LAMBDA (LinkOrLinkIcon Window NewLinkLabel ForceRedisplayFlg) ; Edited 9-Jan-89 10:06 by rtk
; rht: '30-Sep-87 20:52' pp pp
; Relabel a NoteCard link. Ask user for new label. Update all the
; proper references to this link.
```

- ;;; rht 11/19/84: Fixed so that Card is defined before first use.
- ;;; kirk 14Nov85: deleted use of and LinkID
- ;;; kef 8/8/86: Added obtaining write lock and NC.PutFromLinks.
- ;;; fgh 8/30/86 Adapted to NC.IfCardPartNotBusy
- ;;; rht 9/29/86: Changed Ken's call to NC.CardBeingEditedP to NC.ActiveCardP and other minor mod's.
- ;;; rg 5/15/87 now calls NC.LinksCachedP instead of NC.ActiveCardP
- ;;; rht 5/27/87: Now passes new label through cross file link if its dest notefile is open.
- ;;; pmi 9/11/87: Fixes bug 692 --- can no longer change link label when notefile is open read-only.
- ;;; rht 9/30/87: Now tries to update ShowLinks windows if any.
- ;;; bk 1/6/89 Redisplay the Window if a change.

```
(LET (Link LinkIcon Card DestinationCard OldLabel NoteCardType GlobalLinkFlg)
  [if (NC.LinkIconImageObjP LinkOrLinkIcon)
    then (SETQ Link (NC.FetchLinkFromLinkIcon LinkOrLinkIcon))
    (SETQ LinkIcon LinkOrLinkIcon)
```

```

else (SETQ Link LinkOrLinkIcon)
  (OR (SETQ GlobalLinkFlg (NC.GlobalLinkP Link))
      (SETQ LinkIcon (NC.FetchLinkIconForLink Link)
      (SETQ Card (fetch (Link SourceCard) of Link))
      (if (NC.CheckForNotReadOnly Card Window "Can't change link type of links in ")
          then [COND
              ((WINDOWP Window))
              ((NC.ActiveCardP Card)
              (SETQ Window (NC.FetchWindow Card)
              (SETQ DestinationCard (fetch (Link DestinationCard) of Link))
              (NC.IfCardPartNotBusy DestinationCard 'FROMLINKS
              (COND
                  ((NC.SystemLinkLabelP (fetch (Link Label) of Link))
                  (NC.PrintMsg Window T "This is a system maintained pointer." (CHARACTER 13)
                  "You cannot change its label."
                  (CHARACTER 13))
                  (SPAWN.MOUSE)
                  (DISMISS 1500)
                  (NC.ClearMsg Window T))
                  ((SETQ NewLinkLabel (OR NewLinkLabel (NC.AskLinkLabel Window NIL NIL T T)))
                  (SETQ NoteCardType (NC.RetrieveType Card))

```

;;; Put new label in Link Icon or global links list

```

(if GlobalLinkFlg
    then (for GlobalLink in (NC.FetchGlobalLinks Card) when (NC.SameLinkP GlobalLink
                                                                    Link)
          do (replace (Link Label) of GlobalLink with NewLinkLabel)
              (NC.SetLinksDirtyFlg Card T)
              (RETURN))
    else (replace (Link Label) of (NC.FetchLinkFromLinkIcon LinkIcon) with NewLinkLabel)
        (NC.MarkCardDirty Card))

```

;;; Update ToLink list of the SourceCard card

```

(for ToLink in (NC.FetchToLinks Card) when (NC.SameLinkP ToLink Link)
  do (replace (Link Label) of ToLink with NewLinkLabel)
      (NC.SetLinksDirtyFlg Card T)
      (RETURN))

```

;;; Update link icons in any displayed ShowLinks windows.

```

(LET ((ShowLinksWin (NC.FetchShowLinksWindow Card)))
  (if ShowLinksWin
      then (NC.ChangeLinkLabelInShowLinksWin Link ShowLinksWin NewLinkLabel)))
(LET ((ShowLinksWin (NC.FetchShowLinksWindow DestinationCard)))
  (if ShowLinksWin
      then (NC.ChangeLinkLabelInShowLinksWin Link ShowLinksWin NewLinkLabel)))

```

;;; Update FromLink list of DestinationCard card

```

(if (NC.LinksCachedP DestinationCard)
    then (for FromLink in (NC.FetchFromLinks DestinationCard)
          when (NC.SameLinkP FromLink Link)
          do (replace (Link Label) of FromLink with NewLinkLabel)
              (NC.SetLinksDirtyFlg DestinationCard T)
              (RETURN))
    else (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of DestinationCard))
          (NC.GetLinks DestinationCard)
          (for FromLink in (NC.FetchFromLinks DestinationCard)
                when (NC.SameLinkP FromLink Link)
                do (replace (Link Label) of FromLink with NewLinkLabel)
                    (NC.SetLinksDirtyFlg DestinationCard T)
                    (RETURN))
          (NC.PutFromLinks DestinationCard)
          (NC.UncacheLinks DestinationCard)))

```

;;; Pass change through cross-file link to other notefile if open.

```

[if (NC.CrossFileLinkCardP Card)
  then (LET ((RemoteCrossFileLinkCard (NC.FetchRemoteCrossFileLinkCard Card)))
        (AND RemoteCrossFileLinkCard (NC.CheckCrossFileLinkType
                                        RemoteCrossFileLinkCard Card Link))
        (NC.UpdateLinkImages Window (fetch (Link DestinationCard) of Link)))
[if (NC.CrossFileLinkCardP DestinationCard)
  then (LET ((RemoteCrossFileLinkCard (NC.FetchRemoteCrossFileLinkCard
                                        DestinationCard)))
        (AND RemoteCrossFileLinkCard (NC.CheckCrossFileLinkType
                                        RemoteCrossFileLinkCard
                                        DestinationCard Link))
        (NC.UpdateLinkImages Window (fetch (Link DestinationCard) of Link)))

```

;;; Update images in SourceCard window

```

(if (WINDOWP Window)
    then (NC.UpdateLinkImages Window (fetch (Link DestinationCard) of Link)))
(if (AND (NULL ForceRedisplayFlg)
        (NC.TEditBasedP NoteCardType))
    then 'CHANGED

```

else NIL])

(NC.SystemLinkLabelP

[LAMBDA (LinkLabel)

(\* fgh%: "23-Apr-84 23:07")

(\* Return T in LinkLabel is a label for a system maintained link.
NIL otherwise. System maintained links determined by hardwire list for now.)

(MEMB LinkLabel NC.SystemLinkLabels])

(NC.ValidLinkP

[LAMBDA (Link)

(\* rht%: "27-May-87 12:13")

(\* Check Link to see if it is a valid NOTECARDLINK, including check that the NoteCard specified by DESTINATIONID is an
active card.)

(\* \* kirk 15Nov85%: deleted use of DatabaseStream)

(\* \* rht 7/18/86%: Was actually reading in the destination card from notefile! No more.)

(\* \* rht 5/27/87%: Took out checks for valid source and destination cards.)

(LET (DestinationCard)
(AND (type? Link Link)
(NEQ (fetch (Link UID) of Link)
-1]))

(NC.MakeChildLink

[LAMBDA (Child Parent Window)

(\* Randy.Gobbel "18-Dec-86 18:26")

(\* \* Tries to add Child as a child of Parent using either FiledCard or SubBox link as appropriate.
Won't allow if would cause a cycle. Returns new link if successful, else NIL.)

(\* \* kirk 14Nov85%: deleted use of PSA.Database)

(\* \* rg 12/18/86%: Cycle-checking code moved to NC.EnsureNoCycles)

(DECLARE (GLOBALVARS NC.SubBoxLinkLabel NC.FiledCardLinkLabel)
(AND (NC.EnsureNoCycles Parent Child (FUNCTION NC.ChildLinkP
Window)
(NC.MakeFilingLink Parent Child (if (NC.FileBoxP Child T)
then NC.SubBoxLinkLabel
else NC.FiledCardLinkLabel]))

(NC.EnsureNoCycles

[LAMBDA (Parent Child LinkPredicate InterestedWindow)

(\* Randy.Gobbel "18-Dec-86 18:14")

(\* \* Return T if no cycles via specified link predicate, else NIL.)

(COND
([for Link in (NC.FetchToLinks Parent) thereis (AND (APPLY\* LinkPredicate Link)
(NC.SameCardP Child (fetch (Link DestinationCard)
of Link)
(NC.PrintMsg InterestedWindow NIL (NC.RetrieveTitle Child)
" is already a child of "
(NC.RetrieveTitle Parent)
(CHARACTER 13))
NIL)
((AND (NEQ Parent Child)
(NC.NotDaughterP Child Parent LinkPredicate))
T)
(T (NC.PrintMsg InterestedWindow T (NC.RetrieveTitle Child)
" is an ancestor of "
(NC.RetrieveTitle Parent)
(CHARACTER 13))
NIL]))

(NC.SameLinkP

[LAMBDA (Link1 Link2)

(\* rht%: "15-Nov-85 15:45")

(\* Compare two link UIDs)

(NC.SameUIDP (fetch (Link UID) of Link1)
(fetch (Link UID) of Link2])

(NC.SameLinksP

[LAMBDA (LinkListA LinkListB)

(\* rht%: "15-Nov-85 15:39")

(\* \* ARe these two lists of links the same?)

(for LinkA in LinkListA as LinkB in LinkListB always (NC.SameLinkP LinkA LinkB])

**(NC.AddGlobalLinkToCard**

[LAMBDA (WindowOrTextStream)

(\* rht%: "15-Feb-85 18:54")

(\* \* Called from card's menu to add one global link.)

(NC.AddGlobalLinksToCard WindowOrTextStream T])

**(NC.AddGlobalLinksToCard**

[LAMBDA (CardIdentifier SingleLinkFlg)

(\* pmi%: "21-May-87 15:42")

(\* \* Called from card's menu to add some global links.)

(\* \* rht 11/24/85%: Fixed bugs. It was calling NC.CoerceToID.)

(\* \* fgh |6/9/86| Added chack for other interaction operations already in progress.)

(\* \* pmi 12/5/86%: Modified message to NC.SelectNoteCards to mention SHIFT-selection.)

(\* \* pmi 12/12/86%: Removed obsolete ReturnLinksFlg argument in call to NC.SelectNoteCards.)

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

(\* \* rg |3/18/87| added NC.CardSelectionOperation wrapper)

(\* \* rg |3/31/87| fixed bug in NC.ProtectedCardOperation wrapper)

(\* \* pmi 5/21/87%: Changed NC.CardSelectionOperation to NCP.WithLockedCards)

(NC.ProtectedCardOperation (NC.CoerceToCard CardIdentifier)

"Add Global Link(s)" NIL

(NCP.WithLockedCards (LET (Window SourceCard DestinationCards Label)

(SETQ SourceCard (NC.CoerceToCard CardIdentifier))

(SETQ Window (NC.FetchWindow SourceCard))

(SETQ Label (NC.AskLinkLabel Window NIL NIL T T))

(AND Label (if SingleLinkFlg

then (NC.MakeGlobalLink Window Label NIL SourceCard)

(NC.ClearMsg Window T)

else (SETQ DestinationCards

(NC.SelectNoteCards NIL

[FUNCTION (LAMBDA (Card)

(COND

((NEQ Card SourceCard)

T)

(T (NC.PrintMsg Window T "A

Card/Box cannot point to

itself. " (CHARACTER

13)

"Selection ignored."

(CHARACTER 13))

NIL]

NC.SelectingMultipleCardsMenu SourceCard

"Please shift-select the Cards or Boxes to be

pointed to.")

(NC.ClearMsg Window T)

(for DestinationCard in DestinationCards

do (NC.MakeGlobalLink Window Label DestinationCard

SourceCard])

**(NC.ChangeCardTitleFromLinkIcon**

[LAMBDA (LinkIcon Window)

; Edited 3-Dec-87 19:00 by rht:

(\* \* From a link icon sitting in Window, change the title of the link's destination card.)

(\* \* kirk 14Nov85%: deleted use of PSA.Database)

(\* \* fgh |6/4/86| Fixed so that moves card back to original location if moved to be on screen.

This fixes a bug in TEdit that'll mess up the screen if a window is mived under a ImageObj ButtonEventInFn.)

(\* \* kef 7/16/86%: Added mechanism for grabbing the write permission on the TITLE of the destination card.)

(\* \* kef 7/30/86%: Modified to check for Client's concept of whether he owns the write lock or not, thus deciding whether or not to setup the release of the write lock afterwards.)

(\* \* fgh |8/30/86| Adpated to use NC.IfCardPartNotBusy.)

(\* \* rht 11/13/86%: Now checks for crossfile links and bails out.)

(\* \* rht 2/18/87%: Added check for readonly notefile to fix bug %#376.0)

(LET

(Link Card DestinationCard OldTitle NewTitle)

(RESETLST

[RESETSAVE NIL `(MOVEW ,Window ,(LOWERLEFT (WINDOWPROP Window 'REGION)

[if (NC.LinkIconImageObj LinkIcon)

then

```
(SETQ Link (NC.FetchLinkFromLinkIcon LinkIcon))
(SETQ Card (fetch (Link SourceCard) of Link))
(SETQ DestinationCard (fetch (Link DestinationCard) of Link))
(if (NC.CheckForNotReadOnly Card Window "Can't change titles for cards in ")
    then (if (NC.CrossFileLinkCardP DestinationCard)
        then (NC.PrintMsg Window T "Can't change card title from a cross notefile link.")
              (DISMISS 1500)
              (NC.ClearMsg Window T)
              NIL)
        else (NC.IfCardPartNotBusy
              DestinationCard
              'TITLE
              (if (SETQ NewTitle (NC.AskUser (CONCAT "Enter new title for card with title '"
                                                  (NC.RetrieveTitle DestinationCard)
                                                  "'")
                                                  (CHARACTER 13)))
                  "--> "
                  (if (AND (STREQUAL (SETQ OldTitle (NC.RetrieveTitle
                                                              DestinationCard))
                                  "Untitled")
                          (NC.FetchNewCardFlg DestinationCard))
                      then NIL
                      else OldTitle)
                  T Window))
              then (NC.AssignTitle DestinationCard NIL NewTitle)
              'CHANGED]]))
```

**(NC.UncacheLinks**

```
[LAMBDA (Card) (* rht%: "14-Feb-86 15:06")
```

(\* \* Throw away the links cache for this card.)

```
(replace (Card Links) of Card with NIL])
```

**(NC.TraverseLink**

```
[LAMBDA (LinkIcon Window) (* Randy.Gobbel " 4-Nov-87 13:57")
```

(\* \* Follow the link to the card at the other end. For now, amounts to editing the destination card.)

(\* \* pmi&rht 6/19/87%: Now checks for valid destination card before trying to edit it. Prints message to prompt window and changes link icon to "Deleted" if card is invalid.)

(\* \* rg |11/4/87| calls NC.EditNoteCard w/ ReadOnly TRUE if NoteFile is read-only)

```
(DECLARE (GLOBALVARS NC.MsgDelay))
(LET ((Link (NC.FetchLinkFromLinkIcon LinkIcon))
      (InterestedWindow (NC.CoerceToInterestedWindow (MAINWINDOW Window)))
      (Card OldTitle)
      (SETQ Card (fetch (Link DestinationCard) of Link))
      (if (NC.ValidCardP Card)
          then (SETQ OldTitle (NC.RetrieveTitle Card))
          (NC.EditNoteCard Card (fetch (NoteFile ReadOnlyFlg) of (fetch (Card NoteFile) of Card))))
```

(\* \* We're under TEdit, so if title was changed by EditFn, then be sure link icon gets redisplayed.)

```
(if (STREQUAL OldTitle (NC.RetrieveTitle Card))
    then 'DON'T
    else 'CHANGED)
```

else

(\* \* The link is bad - tell the user and change it to "Deleted")

```
(NC.PrintMsg InterestedWindow T "This is a bad link - the destination card is invalid."
  (CHARACTER 13))
(DISSMISS NC.MsgDelay)
(NC.ClearMsg InterestedWindow T)
(NC.DeleteBadLink Link LinkIcon Window])
```

**(NC.CreateLinkIconStrings**

```
[LAMBDA (LinkIconString Icon CrossFileLinkIcon ImageStream) ; Edited 24-Jan-89 15:35 by rtk
```

:: pmi 2/9/87: First written to parse the text for a link icon into multiple lines

:: rht 10/24/87: Removed initial 'Link' arg. It was not being used.

:: pmi 11/3/87: Added CrossFileLinkIcon argument for change to cross-file link icons (adding attached bitmap to right of link icon).

:: pmi 2/1/88: Fixed numerous small problems with calculating where to break the LinkIconString.

:: bk 1/23/89: OR Scale to 1

```
(DECLARE (GLOBALVARS NC.LinkIconMaxWidth NC.LinkIconMultiLineMode NC.LinkIconBorderWidth
  NC.LinkIconSpaceInnerX))
(PROG ((Scale (OR (DSPSCALE NIL ImageStream)
  1))
  (LinkIconLines NIL)
```

```

(ThisString "")
RemainingLineWidth Font ScaledBorderWidth ScaledSpaceInnerX LineWidth StringLength CharsPtr
NumberOfLines StringList String StringWidth Char CharWidth CurrentWidth EndPtr)
(SETQ Font (DSPFONT NIL ImageStream))
;; Make temporaries of scaled vars.
(SETQ ScaledBorderWidth (TIMES Scale NC.LinkIconBorderWidth))
(SETQ ScaledSpaceInnerX (TIMES Scale NC.LinkIconSpaceInnerX))
;; Width of text in link icon
(SETQ LineWidth (DIFFERENCE (TIMES Scale NC.LinkIconMaxWidth)
                             (PLUS ScaledSpaceInnerX ScaledSpaceInnerX ScaledBorderWidth ScaledBorderWidth))
)
[SETQ RemainingLineWidth (DIFFERENCE LineWidth (PLUS (if Icon
                                                       then (DIFFERENCE (TIMES Scale (BITMAPWIDTH
                                                                                               Icon))
                                                                                               ScaledBorderWidth)
                                                       else 0)
              (if CrossFileLinkIcon
                  then (DIFFERENCE (TIMES Scale (BITMAPWIDTH
                                                                                               CrossFileLinkIcon
                                                                                               ))
                                                                                               ScaledBorderWidth)
                  else 0)]
;; Turn the text into a string list of words and spaces
[if NC.LinkIconMultiLineMode
  then ;; Calculate multiple lines of text
    (SETQ StringList (NC.ParseString LinkIconString))
    (SETQ String (CAR StringList))
    (SETQ NumberOfLines 0)
    ;; Determine the line breaks
    [while StringList
      do (if (AND (STREQUAL (SUBSTRING String 1 1)
                          ' " ")
                (STREQUAL ThisString ""))
            then ;; The 'word' is a space or spaces and we are at the beginning of the line. In this case we want to throw
                ;; away the spaces.
                (SETQ StringList (CDR StringList))
                (if StringList
                    then (SETQ String (CAR StringList)))
                else (SETQ StringWidth (STRINGWIDTH String Font))
                    (if (LESSP StringWidth RemainingLineWidth)
                        then ;; The word or spaces will fit on the remainder of the line
                            (SETQ ThisString (CONCAT ThisString String))
                            (SETQ RemainingLineWidth (DIFFERENCE RemainingLineWidth StringWidth))
                            (SETQ StringList (CDR StringList))
                            (if StringList
                                then (SETQ String (CAR StringList))
                                else (SETQ LinkIconLines (NCONC1 LinkIconLines ThisString)))
                            elseif (EQ StringWidth RemainingLineWidth)
                                then ;; The word or spaces will exactly fit on the remainder of the line
                                    (SETQ ThisString (CONCAT ThisString String))
                                    (SETQ LinkIconLines (NCONC1 LinkIconLines ThisString))
                                    (SETQ ThisString "")
                                    (SETQ RemainingLineWidth LineWidth)
                                    (SETQ NumberOfLines (ADD1 NumberOfLines))
                                    (SETQ StringList (CDR StringList))
                                    (if StringList
                                        then (SETQ String (CAR StringList)))
                                    elseif (STREQUAL ThisString "")
                                        then ;; We have a word that is too long for the line, and we are at the beginning of the line
                                            (SETQ CurrentWidth 0)
                                            [SETQ EndPtr (for I from 1 to (NCHARS String)
                                                         do (SETQ Char (SUBSTRING String I I))
                                                            (SETQ CurrentWidth (PLUS CurrentWidth (STRINGWIDTH
                                                                                               Char Font)))
                                                            (if (GREATERP CurrentWidth RemainingLineWidth)
                                                                then (RETURN (SUB1 I))
                                                                else 0)
                                                         )
                                            [if (EQ EndPtr 0)
                                                then ;; This character won't fit on this line, so give up
                                                    (OR LinkIconLines (SETQ LinkIconLines (LIST "")))
                                                    (RETURN)
                                                else (SETQ ThisString (CONCAT ThisString (SUBSTRING String 1 EndPtr))
                                                    (SETQ LinkIconLines (NCONC1 LinkIconLines ThisString))
                                                    (SETQ String (SUBSTRING String (ADD1 EndPtr)))
                                                    (SETQ ThisString "")
                                                    (SETQ RemainingLineWidth LineWidth)
                                                    (SETQ NumberOfLines (ADD1 NumberOfLines))

```

```

elseif (STREQUAL (SUBSTRING String 1 1)
         ' " ")
  then ;; We have spaces at the end of the line which can be thrown away
        (SETQ LinkIconLines (NCONC1 LinkIconLines ThisString))
        (SETQ ThisString "")
        (SETQ RemainingLineWidth LineWidth)
        (SETQ NumberOfLines (ADD1 NumberOfLines))
        (SETQ StringList (CDR StringList))
        (if StringList
         then (SETQ String (CAR StringList)))
  else ;; We have a word that won't fit on the remainder of the line, so we must start it on the next line
        (SETQ LinkIconLines (NCONC1 LinkIconLines ThisString))
        (SETQ ThisString "")
        (SETQ RemainingLineWidth LineWidth)
        (SETQ NumberOfLines (ADD1 NumberOfLines))
elseif (LESSP (STRINGWIDTH LinkIconString Font)
          RemainingLineWidth)
  then ;; Calculate single line of text. The string will fit as is
        (SETQ LinkIconLines (LIST LinkIconString))
  else ;; Calculate single line of text. We have to chop the string down to fit in one line
        (SETQ StringLength (NCHARS LinkIconString))
        (SETQ CharsPtr 1)
        (SETQ LinkIconLines (LIST (while (ILEQ CharsPtr StringLength)
                                       do [if (LESSP (SETQ CharWidth (STRINGWIDTH (NTHCHAR LinkIconString
                                                                                               CharsPtr)
                                                                                               Font))
                                             RemainingLineWidth)
                                           then (SETQ RemainingLineWidth (DIFFERENCE RemainingLineWidth
                                                                                               CharWidth))
                                           (SETQ CharsPtr (ADD1 CharsPtr))
                                           elseif (EQ CharsPtr 1)
                                           then (RETURN "")
                                           else (RETURN (SUBSTRING LinkIconString 1 (SUB1 CharsPtr)
                                                                    finally (RETURN (SUBSTRING LinkIconString 1 (SUB1 CharsPtr)
                                                                                               (RETURN LinkIconLines])

```

(NC.ShrinkIcon [LAMBDA (WINDOW)

```

; Edited 7-Dec-88 14:57 by krivacic
(* LET* ((NC.ShrinkLinkIconBitMap
          (BITMAPCREATE 200 200 1)) (NC.ShrinkLinkIconStream
          (DSPCREATE NC.ShrinkLinkIconBitMap)))
        (DSPFONT NC.LinkIconFont NC.ShrinkLinkIconStream)
        (LET* ((CARD (NC.CoerceToCard WINDOW))
              (LINK (create Link UID_ (NC.MakeUID) SourceCard _ CARD
                              DestinationCard _ CARD AnchorMode _ NIL Label _
                              (fetch (Card Title) of CARD) DisplayMode _
                              (NC.FetchLinkDisplayMode CARD)))
              (IMAGE-OBJ (NC.MakeLinkIcon LINK))
              (IMAGE-BOX (IMAGEBOX IMAGE-BOX
                                   NC.ShrinkLinkIconStream)) (XSIZE
              (fetch (IMAGEBOX XSIZE) of IMAGE-BOX))
              (YSIZE (fetch (IMAGEBOX YSIZE) of IMAGE-BOX))
              (IMAGE-BITMAP (BITMAPCREATE XSIZE YSIZE 1)))
              (LET ((IMAGE-STREAM (DSPCREATE IMAGE-BITMAP))
                  (X-OFFSET 1) (Y-OFFSET 1) (LINE-WIDTH 1))
                (CL:UNWIND-PROTECT (PROGN
                                   (DSPFONT NC.LinkIconFont IMAGE-STREAM)
                                   (NC.DrawLinkOrPointerIcon IMAGE-STREAM IMAGE-BOX
                                   (fetch (Link Label) of LINK) T CARD NIL NIL)
                                   (DRAWLINE 0 0 0 (-
                                   YSIZE Y-OFFSET) LINE-WIDTH
                                   (QUOTE REPLACE) IMAGE-STREAM)
                                   (DRAWLINE 0 0 (-
                                   XSIZE Y-OFFSET) 0 LINE-WIDTH
                                   (QUOTE REPLACE) IMAGE-STREAM)
                                   (DRAWLINE (-
                                   XSIZE X-OFFSET) 0 (-
                                   XSIZE X-OFFSET) (-
                                   YSIZE Y-OFFSET) LINE-WIDTH
                                   (QUOTE REPLACE) IMAGE-STREAM)
                                   (DRAWLINE 0 (-
                                   YSIZE Y-OFFSET) (-
                                   XSIZE X-OFFSET) (-
                                   YSIZE Y-OFFSET) LINE-WIDTH
                                   (QUOTE REPLACE) IMAGE-STREAM)
                                   (SHRINKW WINDOW IMAGE-BITMAP
                                   (GETBOXPOSITION XSIZE YSIZE NIL NIL NIL "Specify Positon
                                   of Icon")) (SETQ GLOBAL-BM IMAGE-BITMAP))
                                   (CLOSEF IMAGE-STREAM))) (CLOSEF
                                   NC.ShrinkLinkIconStream)))

```

```

(DECLARE (GLOBAVARS NC.IconProps))
(LET* [(OldIconTitle (WINDOWPROP W 'NoteCardsIconTitle))

```

```

(Icon (WINDOWPROP W 'ICON))
(Card (NC.CoerceToCard W))
(IconTitle (NC.RetrieveTitle Card))
[CardType (fetch (CardObject Type) of (WINDOWPROP W 'NoteCardObject)
(IconInfo (LISTGET NC.IconProps CardType))
(IconFn (WINDOWPROP W 'ICONFN))
(New-Title (NOT (EQUAL OldIconTitle IconTitle)
[COND
  ((NOT Icon)
(WINDOWPROP W 'ICONPOSITION NIL)
(WINDOWPROP W 'ICONWINDOW NIL)
(WINDOWDELPROP W 'SKETCH.ICON.TITLE)
(WINDOWDELPROP W 'TEDIT.ICON.TITLE NIL)
(WINDOWPROP W 'ICON (SETQ Icon (TITLEDICONW IconInfo IconTitle NC.IconFont
(PPOSITIONP (WINDOWPROP W 'SHRUNKENWINPOS)
T)))
(WINDOWPROP Icon 'COPYBUTTONEVENTFN 'NC.CopyButtonEventFn)
(WINDOWPROP Icon 'BUTTONEVENTFN 'NC.ShrukenIconButtonEventFn)
(* AND IconFn (APPLY* IconFn W))
]
(COND
(New-Title (WINDOWPROP W 'NoteCardsIconTitle IconTitle)
(ICONTITLE IconTitle (fetch (TITLEDICON TITLEREG) of IconInfo)
NIL Icon))]
)

```

(DEFINEQ

**(NC.AddLinkToCard**

```
[LAMBDA (CardIdentifier LinkOrLinkLabel DestinationCard) (* Randy.Gobbel " 4-Mar-87 14:15")
```

(\* 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)

```

(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 'Text)
(NC.InsertLinkInText (NC.FetchSubstance Card)
LinkOrLinkLabel DestinationCard))
((NC.IsSubTypeOfP CardType 'Graph)
(NC.AddLinkToGraphCard Window LinkOrLinkLabel DestinationCard))
((NC.IsSubTypeOfP CardType 'Sketch)
(NC.AddLinkToSketchCard Window LinkOrLinkLabel DestinationCard)]))
)

```

**(NC.AddLinksToCard**

```
[LAMBDA (CardIdentifier LinkLabel DestinationCards) (* Randy.Gobbel " 2-Apr-87 15:38")
```

(\* Adds multiple links to card designated by CardIdentifier. Rest of args can be NIL.)

(\* fgh |6/9/86| Added Checks to make sure other interaction operation not in progress.)

(\* rht 9/23/86%: Fixed bug in function passed to NC.SelectNoteCards.)

(\* pmi 12/5/86%: Modified message to NC.SelectNoteCards to mention SHIFT-selection.)

(\* pmi 12/12/86%: Removed obsolete ReturnLinksFlg argument in call to NC.SelectNoteCards.)

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

(\* rg |3/18/87| added NC.CardSelectionOperation wrapper)

(\* RG |4/2/87| changed NC.CardSelectionOperation to NCP.WithLockedCards)

```

(LET ((SourceCard (NC.CoerceToCard CardIdentifier))
CardType Window OperationInProgress)
(NC.ProtectedCardOperation SourceCard "Insert Links" NIL
(NCP.WithLockedCards (SETQ Window (NC.FetchWindow SourceCard))
(SETQ CardType (NC.FetchType SourceCard))
(OR LinkLabel (SETQ LinkLabel (NC.AskLinkLabel Window NIL NIL T T)))
(if LinkLabel
then (OR DestinationCards (SETQ DestinationCards
(NC.SelectNoteCards NIL [FUNCTION (LAMBDA (SelectedCard)
(COND
((NOT (NC.SameCardP
SelectedCard
SourceCard))

```





then (replace (Link Label) of FoundLink with NewLinkLabel))

)

::: New Display Mode format

(DEFINEQ

**(NC.CheckDisplayModeFormat**

[LAMBDA (Link) (\* fgh%: "16-Nov-85 17:56")

(\* If Link's display mode is an atom, then change to the new record format.)  
(\* rht 3/28/85%: Now does nothing if link is not in current format.  
In that case, we should be inside of a ConvertVersion0ToVersion1 so that a later repair should fix the links.)  
(\* rht 7/17/85%: Now turns into a typed record if wasn't already.)  
(\* fgh |11/16/85| Removed conversion to typed record, since links are now datatypes.)

(LET (DisplayMode)  
[COND  
((type? Link Link)  
(COND  
((ATOM (SETQ DisplayMode (fetch (Link DisplayMode) of Link)))  
(replace (Link DisplayMode) of Link with (NC.MakeNewDisplayMode DisplayMode)))  
((NEQ (CAR DisplayMode)  
'LINKDISPLAYMODE)  
(replace (Link DisplayMode) of Link with (CONS 'LINKDISPLAYMODE DisplayMode]  
Link])

**(NC.InsureLinkDisplayMode**

[LAMBDA (LinkDisplayMode) (\* rht%: " 6-Oct-86 23:40")

(\* Return the recordized form of the LinkDisplayMode argument.)  
(\* fgh |5/2/86| First created)  
(\* rht 10/6/86%: Now uses FLOAT for ATTACHBITMAPFLG in most cases where LinkDisplayMode is atomic.)

(COND  
((type? LINKDISPLAYMODE LinkDisplayMode)  
LinkDisplayMode)  
(T (SELECTQ LinkDisplayMode  
(Title (create LINKDISPLAYMODE  
SHOWTITLEFLG \_ T  
ATTACHBITMAPFLG \_ 'FLOAT))  
(Icon (create LINKDISPLAYMODE  
ATTACHBITMAPFLG \_ T))  
(Label (create LINKDISPLAYMODE  
SHOWLINKTYPEFLG \_ T  
ATTACHBITMAPFLG \_ 'FLOAT))  
(Both (create LINKDISPLAYMODE  
SHOWTITLEFLG \_ T  
SHOWLINKTYPEFLG \_ T  
ATTACHBITMAPFLG \_ 'FLOAT))  
(create LINKDISPLAYMODE  
ATTACHBITMAPFLG \_ T]))

**(NC.MakeNewDisplayMode**

[LAMBDA (ModeAtom) (\* rht%: " 9-Sep-85 22:12")

(\* Makes a new display mode record structure using info in ModeAtom.)  
(create LINKDISPLAYMODE  
SHOWTITLEFLG \_ (AND (FMEMB ModeAtom '(Title Both))  
T)  
SHOWLINKTYPEFLG \_ (AND (FMEMB ModeAtom '(Label Both))  
T)  
ATTACHBITMAPFLG \_ (if (OR (NULL ModeAtom)  
(EQ ModeAtom 'Icon))  
then T  
else 'FLOAT])

**(NC.DisplayModeFromStylesheetSelections**

[LAMBDA (Selections) (\* rht%: "18-Jul-85 14:35")

(\* Return the display mode corresponding to the user's selection from the changeDisplayMode stylesheet.)

(AND Selections (create LINKDISPLAYMODE  
SHOWTITLEFLG \_ (LISTGET NC.MenuItems/DisplayModeValues (CAR Selections))  
SHOWLINKTYPEFLG \_ (LISTGET NC.MenuItems/DisplayModeValues (CADR Selections))  
ATTACHBITMAPFLG \_ (LISTGET NC.MenuItems/DisplayModeValues (CADDR Selections))

**(NC.StylesheetSelectionsFromDisplayMode**

[LAMBDA (DisplayMode)

; Edited 9-Jan-89 10:20 by rtk

;;; BK 1/9/89Change FLOAT Names to System Defaults

;;; Return the list of ChangeDisplayMode stylesheet selections corresponding to the given DisplayMode.

```
(LIST (SELECTQ (fetch (LINKDISPLAYMODE SHOWTITLEFLG) of DisplayMode)
      (NIL 'No)
      (FLOAT 'System% Defaults)
      'Yes)
      (SELECTQ (fetch (LINKDISPLAYMODE SHOWLINKTYPEFLG) of DisplayMode)
      (NIL 'No)
      (FLOAT 'System% Defaults)
      'Yes)
      (SELECTQ (fetch (LINKDISPLAYMODE ATTACHBITMAPFLG) of DisplayMode)
      (NIL 'No)
      (FLOAT 'System% Defaults)
      'Yes])
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.MenuItems/DisplayModeValues)
)

(RPAQ? NC.MenuItems/DisplayModeValues '(Yes T No NIL System% Defaults FLOAT Yes T No NIL System% Defaults FLOAT
Yes T No NIL System% Defaults FLOAT))
```

;;; New fns to support No-Link cards/substances

(DEFINEQ

**(NC.LinksSupportedP**

[LAMBDA (Card AnchorModesList)

(\* fgh%: " 5-Feb-86 14:08")

(\* \* Are all of the anchor modes supported by the card Card)

(\* \* fgh |2/5/86| Changed call to LinkAnchorModesFromType to NC.FetchLinkAnchorModesSupported)

```
(LET ((SupportedAnchorModes (NC.FetchLinkAnchorModesSupported Card)))
(COND
  ((EQ SupportedAnchorModes T))
  ((NULL SupportedAnchorModes)
   NIL)
  (T (SETQ SupportedAnchorModes (MKLIST SupportedAnchorModes))
     (for AnchorMode in AnchorModesList always (FMEMB AnchorMode SupportedAnchorModes)))
))
```

;;; Link display icons

(DEFINEQ

**(NC.FetchLinkIconForLink**

[LAMBDA (Link)

(\* fgh%: " 5-Feb-86 19:52")

(\* \* Find the Link Icon corresponding to Link)

(\* \* kirk 14Nov85%: deleted use of LinkID and PSA.Database)

(\* \* fgh |2/5/86| Added call to NC.ApplyFn)

```
(LET ((SourceCard (fetch (Link SourceCard) of Link)))
(for LinkIcon in (CAR (NC.ApplyFn CollectLinksFn SourceCard NIL T))
thereis (if (NC.SameLinkP Link (NC.FetchLinkFromLinkIcon LinkIcon))
then LinkIcon])
```

**(NC.MakeLinkIcon**

[LAMBDA (Link)

(\* fgh%: " 1-Oct-84 20:48")

(\* \* Create a Link Icon image object from the lionk info in Link)

```
(IMAGEOBJCREATE Link NC.LinkIconImageFns])
```

**(NC.LinkIconButtonEventInFn**

[LAMBDA (ImageObject Window)

; Edited 10-Mar-89 16:55 by krivacic  
; If the guy buttons inside a link image object

;;; bk 12/14/88: Start the button event fns as separate processes so Shift Select from a TEdit will work

;;; fgh 6/3/86 Removed hot region, removed push copy link icon.

;;; fgh 6/29/86 Put back in a hot region.

;;; rht 11/18/86: Now calls NC.LinkIconLeftButtonFn rather than hardwiring in the call to NC.EditNoteCard.

;;; rht 10/25/87: Now handles Pointer icons as well as link icons.

;;; bk 3/10/89: Remove the ADD.PROCESS for left buttons.

```
(LET ((Window (COND
  ((WINDOWP Window)
   Window)
  ((DISPLAYSTREAMP Window)
   (WFROMDS Window))
  ((TEXTSTREAMP Window)
   (NC.TEditWindow Window))
  (T NIL)))
  (ClippingRegion (DSPCLIPPINGREGION NIL Window))
  HotRegion MouseButtonState TTYWindow ClippingRegionWidth)
```

;;; Figure out the hot region -- basically the clipping region minus a few pts at right and left to allow for TEdit selection of icon.

```
[SETQ HotRegion (if (GREATERP (SETQ ClippingRegionWidth (fetch (REGION WIDTH) of ClippingRegion))
  24)
  then ; There's at least 10 pts between the two 7pt cold regions at left
        ; and right.
        (CREATEREGION (PLUS (fetch (REGION LEFT) of ClippingRegion)
  7)
  (fetch (REGION BOTTOM) of ClippingRegion)
  (DIFFERENCE (fetch (REGION WIDTH) of ClippingRegion)
  14)
  (fetch (REGION HEIGHT) of ClippingRegion))
  elseif (AND (LESSP ClippingRegionWidth 9)) ; if link icon width less than 9 then no cold regions
  then ClippingRegion
  else ; Divide into 4ths and give 1/4 at each edge as cold regions.
        (CREATEREGION (PLUS (fetch (REGION LEFT) of ClippingRegion)
  (TIMES 0.25 ClippingRegionWidth))
  (fetch (REGION BOTTOM) of ClippingRegion)
  (QUOTIENT ClippingRegionWidth 2)
  (fetch (REGION HEIGHT) of ClippingRegion])
(COND
  ((NOT (INSIDEP HotRegion (LASTMOUSEX Window)
  (LASTMOUSEY Window)))
```

;;; If in the link icon, but not in the hot region -- want to select the link icon.

```
(NIL)
(T
```

;;; Otherwise, invert the window and wait until the mouse is up or the cursor moves out of the link icon.

```
(RESETLST
  (RESETSAVE (INVERTW Window))
  (until [OR (MOUSESTATE UP)
  (NOT (INSIDEP HotRegion (LASTMOUSEX Window)
  (LASTMOUSEY Window))]
  do (SETQ MouseButtonState LASTMOUSEBUTTONS)))
```

;;; If user let up inside the link icon, then call the appropriate function.

```
(COND
  ((AND (INSIDEP HotRegion (LASTMOUSEX Window)
  (LASTMOUSEY Window))
  MouseButtonState)
```

;;; User let up inside the link icon.

```
(COND
  ((OR (KEYDOWNP 'LSHIFT)
  (KEYDOWNP 'RSHIFT)
  (KEYDOWNP 'CTRL)
  (KEYDOWNP 'COPY)
  (KEYDOWNP 'DELETE)
  (KEYDOWNP 'MOVE)) ; Copy or move or delete key is down, just do selection
  NIL)
  [(ZEROP (LOGXOR MouseButtonState 4)) ; The left button was down, call the left button fn.
  (COND
    ((NC.LinkIconImageObjP ImageObject)
     (NC.LinkIconLeftButtonFn ImageObject Window))
    ((NC.PointerIconImageObjP ImageObject)
     (NC.PointerIconLeftButtonFn ImageObject Window))]
  [(OR (ZEROP (LOGXOR MouseButtonState 1))
  (ZEROP (LOGXOR MouseButtonState 6))) ; The middle button was down, call the middle button fn.
  (COND
```

```

    [(NC.LinkIconImageObjP ImageObject)
     (ADD.PROCESS `(NC.LinkIconMiddleButtonFn ,ImageObject ,Window)
      (NC.PointerIconImageObjP ImageObject)
      (ADD.PROCESS `(NC.PointerIconMiddleButtonFn ,ImageObject ,Window)
        ((ZEROP (LOGXOR MouseButtonState 2)) ; The right button was down, allow selection
         NIL)
        (T ; Hmmm, don't know know, allow selectin
         NIL)))
     ([AND (INSIDEP ClippingRegion (LASTMOUSEX Window)
              (LASTMOUSEY Window))
          (NOT (INSIDEP HotRegion (LASTMOUSEX Window)
              (LASTMOUSEY Window))

```

;;; User let up outside the hot region but inside the link icon -- select the link icon

```

NIL)
(T

```

;;; User let up outside the link icon -- don't select the link icon

```

'DON'T])

```

**(NC.LinkIconDisplayFn**

[LAMBDA (ImageObj ImageStream StreamType TextStream Scale) ; Edited 21-Jan-88 18:23 by Trigg

;;; Display a link icon

```

;; rht 11/13/84: Made width of box lines also scale dependent.
;; rht 12/4/84: Hacked so type-dependent icons come out optionally to left of text.
;; rht 2/9/85: Changed to use new display mode format.
;; fgh 2/5/86 Added call to NC.FetchLinkIconAttachedBitmap
;; rht & fgh 5/9/86: Massive hacking to display coords, etc.
;; fgh 5/9/86 Added DSPFONT kludge after Typelcon BITBLT to get around bug in PRESS BITBLT. Bug reported as AR #5630.0
;; rht 5/10/86: Rearranged order of expressions a bit and added arg to call to NC.FetchLinkIconAttachedBitmap in order to get a bitmap with correct
;; height for the box we're drawing.
;; rht 8/7/86: Now passes Scale argument to NC.FetchLinkIconAttachedBitmap. Also uses the Title and Label args if non-nil. If not, then
;; recomputes them more sensibly.
;; rht 8/25/86: Fixed improperly placed comment.
;; pmi 2/11/87: Overhauled to accommodate multi-line link icons
;; pmi 2/27/87: Fixed problems with only displaying the attached bitmap and with displaying null titles
;; pmi 5/1/87: added missing local vars declarations.
;; pmi&rht 6/19/87: Added check for bad link (ie: half link to a deleted card.) Such a link will now be deleted and its image object replaced with a
;; 'Bad Link' object.
;; rht 10/24/87: Ripped out heart of this function and moved to NC.DrawLinkOrPointerIcon. In the case when there is no stored ImageBox on the
;; Imageobj, it was doing the calculation twice, once here and then again when the ImageBoxFn is called. Now just calls the ImageBoxFn if
;; necessary up front.
;; rht 1/21/88: Now passes extra BorderWidthIncrement argument to NC.DrawLinkOrPointerIcon which it grabs off IMAGEOBJPROP.

```

```

(DECLARE (GLOBALVARS NC.LinkIconAttachBitmapFlg NC.LinkIconShowLinkTypeFlg NC.LinkIconShowTitleFlg))
(RESETLST

```

```

[RESETSAVE NIL `(DSPFONT , (DSPFONT NC.LinkIconFont ImageStream)
,ImageStream)
[PROG ((Scale (DSPSCALE NIL ImageStream))
(Link (NC.FetchLinkFromLinkIcon ImageObj))
ShowTitleFlg ShowLinkTypeFlg AttachBitmapFlg DisplayType Card Title Label LinkIconString)

```

;; Check for bad Link

```

(if [NOT (AND (NC.ValidCardP (fetch (Link SourceCard) of Link))
              (NC.ValidCardP (fetch (Link DestinationCard) of Link))
      then ;; Delete the offending link, smash its image object into a 'Deleted' image object, and return by displaying it.
      (NC.DeleteBadLink Link ImageObj ImageStream StreamType TextStream Scale)
      (RETURN))

```

;; Link is okay, continue

;; Determine what type of Display to do

```

(SETQ DisplayType (fetch (Link DisplayMode) of Link))
(SETQ ShowTitleFlg (fetch (LINKDISPLAYMODE SHOWTITLEFLG) of DisplayType))
(SETQ Card (if (EQ ShowTitleFlg 'SOURCE)
              then (fetch (Link SourceCard) of Link)
              else (fetch (Link DestinationCard) of Link)))
(SETQ ShowLinkTypeFlg (fetch (LINKDISPLAYMODE SHOWLINKTYPEFLG) of DisplayType))
(SETQ AttachBitmapFlg (fetch (LINKDISPLAYMODE ATTACHBITMAPFLG) of DisplayType))
(if (EQ AttachBitmapFlg 'FLOAT)
    then (SETQ AttachBitmapFlg NC.LinkIconAttachBitmapFlg))

```

;; Construct the text for the link icon

```

(SETQ Title (if (AND ShowTitleFlg (OR (NEQ ShowTitleFlg 'FLOAT)
                                      NC.LinkIconShowTitleFlg))
               then (NC.RetrieveTitle Card)

```

```

        else NIL))
    (SETQ Label (AND (COND
        ((EQ ShowLinkTypeFlg 'FLOAT)
         NC.LinkIconShowLinkTypeFlg)
        (T ShowLinkTypeFlg))
        (fetch (Link Label) of Link)))
    (SETQ LinkIconString (CONCAT (COND
        (Label (CONCAT "<" Label ">"))
        (T ""))
        (COND
        ((AND Label Title)
         " ")
        (T ""))
        (OR Title "")))

```

;; Now do all the hard work.

```

    (NC.DrawLinkOrPointerIcon ImageStream (OR (IMAGEOBJPROP ImageObj 'BOUNDBOX)
        (NC.LinkIconImageBoxFn ImageObj ImageStream NIL NIL NIL
         DisplayType Title Label LinkIconString))
     LinkIconString AttachBitmapFlg Card NIL (IMAGEOBJPROP ImageObj 'BorderWidthIncrement]))

```

**(NC.LinkIconImageBoxFn**

```

[LAMBDA (ImageObj ImageStream CurrentX RightMargin DummyArg DisplayType Title Label LinkIconString)
; Edited 24-Jan-89 15:54 by rtk

```

;; rht 9/20/84: Now scales result before returning by proper amount depending on stream type. e.g. for PRESS and INTERPRESS.  
 ;; rht 11/13/84: In computation of XSIZE, extra width is figured using characters in the font, 'nn' (\, rather) than absolute pixel count.  
 ;; rht 2/9/85: Now uses new displaymode format.  
 ;; rht 5/9/86: Note that RightMargin non-nil signals a TEdit stream.  
 ;; rht & fgh 5/9/86: Massive hacking to display coords, etc.  
 ;; rht 5/10/86: Rearranged order of expressions a bit and added arg to call to NC.FetchLinkIconAttachedBitMap in order to get a bitmap with correct  
 ;; height for the box we're drawing.  
 ;; rht 8/7/86: Now passes Scale argument to NC.FetchLinkIconAttachedBitMap. Also uses the Title and Label args if non-nil. If not, then  
 ;; recomputes them more sensibly.  
 ;; rht 8/25/85: Fixed improperly placed comment.  
 ;; rg 11/18/86: Added TotalEdgeSpaceY to local vars declaration  
 ;; pmi 2/6/87: Overhauled to accommodate multi-line link icons  
 ;; pmi 2/27/87: Fixed problems with only displaying the attached bitmap and with displaying null titles  
 ;; pmi 5/1/87: added missing local vars declarations.  
 ;; pmi&rg 7/15/87 recreated peggy's patch to deal w/ links to deleted cards  
 ;; rht 10/24/87: Ripped out heart of code and put in NC.ComputeLinkOrPointerImageBox. Removed Icon and LinkIconStrings args as both were  
 ;; being computed here anyway.  
 ;; rht 1/21/88: Now passes extra BorderWidthIncrement argument to NC.ComputeLinkOrPointerImageBox which it grabs off IMAGEOBJPROP.

;;; bk 1/23/89: Add CurrentX to call of NC.ComputeLinkOrPointerImageBox.

```

(DECLARE (GLOBALVARS NC.LinkIconAttachBitmapFlg NC.LinkIconShowLinkTypeFlg NC.LinkIconShowTitleFlg))
(RESETLST

```

```

[RESETSAVE NIL `(DSPFONT , (DSPFONT NC.LinkIconFont ImageStream)
, ImageStream)

```

```

(PROG ((Link (NC.FetchLinkFromLinkIcon ImageObj))
Card ShowTitleFlg AttachBitmapFlg ShowLinkTypeFlg)

```

;; Check for bad Link

```

    (if [NOT (AND (NC.ValidCardP (fetch (Link SourceCard) of Link))
        (NC.ValidCardP (fetch (Link DestinationCard) of Link))
        then ;; Delete the offending link, smash its image object into a 'Deleted' image object, and return by displaying it.
        (RETURN (NC.DeletedLinkImageBoxFn ImageObj ImageStream)))

```

;; Determine what type of Display to do

```

    (OR DisplayType (SETQ DisplayType (fetch (Link DisplayMode) of Link)))
    (SETQ ShowTitleFlg (fetch (LINKDISPLAYMODE SHOWTITLEFLG) of DisplayType))
    (SETQ Card (if (EQ ShowTitleFlg 'SOURCE)
        then (fetch (Link SourceCard) of Link)
        else (fetch (Link DestinationCard) of Link)))
    (SETQ ShowLinkTypeFlg (fetch (LINKDISPLAYMODE SHOWLINKTYPEFLG) of DisplayType))
    (SETQ AttachBitmapFlg (fetch (LINKDISPLAYMODE ATTACHBITMAPFLG) of DisplayType))
    (SETQ AttachBitmapFlg (if (EQ AttachBitmapFlg 'FLOAT)
        then NC.LinkIconAttachBitmapFlg
        else AttachBitmapFlg))

```

;; Construct the text for the link icon

```

    (if (NOT LinkIconString)
        then (OR Title (SETQ Title (if (AND ShowTitleFlg (OR (NEQ ShowTitleFlg 'FLOAT)
            NC.LinkIconShowTitleFlg))
            then (NC.RetrieveTitle Card)
            else NIL)))
        [OR Label (SETQ Label (AND (if (EQ ShowLinkTypeFlg 'FLOAT)
            then NC.LinkIconShowLinkTypeFlg

```

```

                else ShowLinkTypeFlg)
            (fetch (Link Label) of Link]
    (SETQ LinkIconString (CONCAT (COND
        (Label (CONCAT "<" Label ">"))
        (T ""))
        (COND
            ((AND Label Title)
             " ")
            (T ""))
        (OR Title ""))

```

;; Now do the hard work.

```

    (RETURN (NC.ComputeLinkOrPointerImageBox ImageStream LinkIconString AttachBitmapFlg Card RightMargin
        (IMAGEOBJPROP ImageObj 'BorderWidthIncrement)
        CurrentX))))

```

**(NC.LinkIconImageObjP**

```

[LAMBDA (IMAGEOBJ) (* fgh%: " 1-Oct-84 20:04")

```

(\* \* Is IMAGEOBJ a Link Icon?)

```

(AND (IMAGEOBJP IMAGEOBJ)
    (EQ (FUNCTION NC.LinkIconGetFn)
        (IMAGEOBJPROP IMAGEOBJ 'GETFN))

```

**(NC.LinkIconPutFn**

```

[LAMBDA (ImageObject Stream) (* rht%: "15-Jul-86 14:52")

```

(\* \* Writes the Link information in the link icon specified by ImageObject onto the file specified by Stream)

(\* \* fgh |11/16/85| Updated to handle Link DATATYPE)

(\* \* rht 7/15/86%: Now checks for old style links. If so, then put down a placeholder imageobj.)

```

(LET ((Link (NC.FetchLinkFromLinkIcon ImageObject)))
    (if (type? Link Link)
        then (NC.WriteLink Link Stream)
        else (IMAGEOBJPROP ImageObject 'OBJECTDATUM "[[Old style link.]]")
            (replace (IMAGEOBJ IMAGEOBJFNS) of ImageObject with NC.ExternalPutLinkIconImageFns)
            (NC.ExternalPutLinkIconPutFn ImageObject Stream))

```

**(NC.LinkIconGetFn**

```

[LAMBDA (Stream) (* fgh%: "16-Nov-85 20:40")

```

(\* \* Reads the link information from Stream and returns a link icon image object for this link info.  
If the value on the stream is an atom in the old style, convert to a proper record.)

```

(NC.MakeLinkIcon (NC.ReadLink Stream))

```

**(NC.LinkIconMiddleButtonFn**

```

[LAMBDA (LinkIcon Window) ; Edited 23-Jan-88 20:44 by Trigg

```

;; Middle button has been pressed in the Notecard link icon. Bring up menu of choices for manipulating this link icon.

;; rht 11/18/86: Now calls NC.TraverseLink rather than NC.EditNoteCard.

;; rht 2/4/87: Now checks the ExtraLinkIconMenuItems property of the destination's card type atom. If non-nil, then include those items in link icon menu.

;; pmi 3/25/87: Added NC.MenuFont to all menus

;; rht 1/16/88: If there are extra items for the menu, then add dashed line boundary to menu. Also now caches middle button menu on card type atom in the extra items case.

;; rht 1/23/88: Now passes Window into the apply\* for any extra link icon menu items.

```

(DECLARE (GLOBALVARS NC.LinkIconMiddleButtonMenu NC.MenuFont))
(LET ((Card (fetch (Link DestinationCard) of (NC.FetchLinkFromLinkIcon LinkIcon)))
    ExtraItems CardType Selection Menu CachedMenu)
    (SETQ CardType (NC.RetrieveType Card))
    (SETQ CachedMenu (GETPROP CardType 'LinkIconMiddleButtonMenu))
    (SETQ ExtraItems (GETPROP CardType 'ExtraLinkIconMenuItems))
    [COND
        ((type? MENU CachedMenu)
         (SETQ Menu CachedMenu))
        ((AND (BOUNDP 'NC.LinkIconMiddleButtonMenu)
              (type? MENU NC.LinkIconMiddleButtonMenu)
              (NULL ExtraItems))
         (SETQ Menu NC.LinkIconMiddleButtonMenu))
        (T (SETQ Menu (create MENU
            ITEMS _ (APPEND '(|Bring Up Card/Box| 'Edit "Bring up the note card pointed to
                by this pointer.")
                (|Change Link Type| 'Relabel "Change the label of this
                pointer.")
                (|Change Card Title| 'Retitle "Change the destination card's
                title.")
                (|Change Display Mode| 'Display "Change the type of

```

```

                                information displayed in this pointer.))
                                [if ExtraItems
                                  then ' (----- NIL "")
                                ExtraItems)
                                CENTERFLG _ T
                                TITLE _ "Link Ops"
                                MENUFONT _ NC.MenuFont
                                ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                                  1))
                                (if ExtraItems
                                  then (PUTPROP CardType 'LinkIconMiddleButtonMenu Menu)
                                  else (SETQ NC.LinkIconMiddleButtonMenu Menu)
                                (if (NC.CardP (NC.CoerceToCard Window))
                                  then (SELECTQ (SETQ Selection (MENU Menu))
                                                (Edit (NC.TraverseLink LinkIcon Window))
                                                (Display (NC.ChangeLinkDisplayMode LinkIcon Window))
                                                (Relabel (NC.RelabelLink LinkIcon Window))
                                                (Retitle (NC.ChangeCardTitleFromLinkIcon LinkIcon Window))
                                                (NIL NIL)
                                                (ADD.PROCESS `(APPLY* ', Selection ', Card ', Window)
                                                             'NAME
                                                             'LinkIconMiddleButton))
                                  else NIL))

```

**(NC.LinkIconWhenCopiedFn**

```
[LAMBDA (ImageObject ToWindowStream FromTextStream ToTextStream)
```

; Edited 15-Feb-89 16:50 by krivacic

```

;; Called when copying a link icon from FromTextStream to ToWindowStream. Sets the necessary link information up for card corresponding to
;; ToWindowStream.
;; rht 11/18/84: Major hacking. Now checks for all sorts of illegal cases. Either goes ahead with copy, converts link type to 'Unspecified' (\, or)
;; deletes the new 'invisible' link. The latter will change when imageobj fns can return DON'T.
;; rht 12/12/84: now does RETFROM instead of adding processes to delete imageobj's. This is cleaner, but still ugly.
;; rht 3/25/85: Fixed to handle copies within a sketch card. That is indicated by nullity of the two textstream args.
;; rht 3/24/86: Changed call to NC.CoerceToID to NC.CoerceToCard
;; fgh 6/30/86 Now ERROR! out of attempt to copy between NoteFiles
;; rht 7/14/86: Reversed order of tests for WINDOWP and TEXTSTREAMP in computation of NewSourceCard.
;; rht 8/11/86: Now only resets NewSourceCard if it was NIL so that copies to a sketch will work.
;; rht 10/5/86: Undid change fgh 6/30/86: now allows copies of links across notefiles. Also handles illegal copies differently. Rather than
;; RETFROM'ing or ERROR!'ing, it puts in a DELETE ME imageobj.
;; rht 12/16/86: Now passes Window to NC.GetCrossFileLinkDestCard.
;; rht 1/21/87: Now handles attempt to copy to non-NoteCards stream differently than other illegalities.
;; rht 1/25/87: Now stashes new link created on source card's user data props for use by NC.LinkAtCharPos in case this is a copy of multiple links.
;; rht 6/24/87: Undid change of 3/25/85. No longer needed and causes bugs.
;; rht 7/1/87: Removed action of changing linkdisplaymode when copying system links.
;; rht 1/8/88: Added new condition to disallow links from cards to themselves.
;; bk 2/15/89: Use NC.REPLACEWITHDELETEDLINKIMAGEOBJ when IllegalCopyFlg is set to smash the ImageObjectCopy object.

```

```

(DECLARE (GLOBALVARS NC.CopyCrossFileLinksMode NC.DELETEMEImageObjDatum NC.SystemLinkLabels
                  NC.UnspecifiedLinkLabel NC.NewCrossFileLinksTwoWayFlg))
(LET ((Link (NC.FetchLinkFromLinkIcon ImageObject))
      (ImageObjectCopy (IMAGEOBJPROP ImageObject 'ImageObjectCopy))
      (InTEditCopyFlg (COND
                       ((STKPOS 'TEDIT.COPY)
                        T)))
      (Window (AND ToWindowStream (WFROMMS ToWindowStream T)))
      SourceCard DestinationCard NewSourceCard NewLabel OldDisplayMode Label InsertPos IllegalCopyFlg
      ExternalCopyFlg)
      (SETQ Label (fetch (Link Label) of Link))
      (SETQ SourceCard (fetch (Link SourceCard) of Link))
      (SETQ DestinationCard (fetch (Link DestinationCard) of Link))
      (SETQ OldDisplayMode (fetch (Link DisplayMode) of Link))
      [SETQ NewSourceCard (COND
                          ((TEXTSTREAMP ToTextStream)
                           (NC.CoerceToCard ToTextStream))
                          ((WINDOWP Window)
                           (NC.CoerceToCard Window))
                          [COND
                           ((TEXTSTREAMP ToTextStream)
                            ; If it's a text copy, then compute position to insert link at.
                            (SETQ InsertPos (NC.CharPosFromTextStream ToTextStream))
                           [COND
                            ((NULL NewSourceCard)
                             ; Trying to copy to a non NoteCard stream
                             (NC.PrintMsg NIL NIL (CONCAT "Tried to copy a NoteCards link icon" " to a non-NoteCards stream."
                                                         (CHARACTER 13)))
                             (SETQ ExternalCopyFlg T))
                            ([NULL (NC.LinksSupportedP NewSourceCard '(Local))
                             (NC.PrintMsg Window NIL (CONCAT "Tried to copy a NoteCards link icon" " to a NoteCard that" " does
                                                             not support links!." (CHARACTER 13)))
                             (SETQ IllegalCopyFlg T))
                             ((NC.SameCardP NewSourceCard DestinationCard)

```



```

(NC.PrintMsg Window NIL (CONCAT "Sorry, links aren't allowed to point from cards to themselves."
                                (CHARACTER 13)))
(SETQ IllegalCopyFlg T)
((AND (FMEMB Label '(FiledCard SubBox))
      (NEQ (NC.RetrieveType NewSourceCard)
           'FileBox))
      ; Copy from filebox to non-filebox.
(NC.PrintMsg Window NIL (CONCAT "Tried to copy filecard or subbox link to a non-filebox."
                                (CHARACTER 13)
                                "Link type of copy set to 'Unspecified'."
                                (CHARACTER 13)))
(SETQ NewLabel NC.UnspecifiedLinkLabel))
((AND (NEQ (NC.RetrieveType SourceCard)
           'FileBox)
      (EQ (NC.RetrieveType NewSourceCard)
          'FileBox))
      ; Copy from non-filebox to filebox.
(NC.PrintMsg Window NIL (CONCAT "Can't copy links from non-filebox to filebox." (CHARACTER 13)
                                "Try using 'PutCardsHere'."
                                (CHARACTER 13)))
(SETQ IllegalCopyFlg T))
((AND (FMEMB Label '(FiledCard SubBox))
      (EQ NewSourceCard SourceCard))
      ; Copy within same filebox.
(NC.PrintMsg Window NIL (CONCAT "Can't copy links within a FileBox. Try move instead."
                                (CHARACTER 13)))
(SETQ IllegalCopyFlg T))
([AND (EQ (NC.RetrieveType NewSourceCard)
          'FileBox)
      (for Link1 in (NC.FetchToLinks NewSourceCard) thereis (AND (NC.ChildLinkP Link1)
                                                                (EQ DestinationCard
                                                                 (fetch (Link DestinationCard)
                                                                    of Link1))
                                                                ; Copy to a filebox already containing this child.
(NC.PrintMsg Window NIL (CONCAT (NC.RetrieveTitle DestinationCard)
                                " not copied: already appears as a child of "
                                (NC.RetrieveTitle NewSourceCard)
                                (CHARACTER 13)))
(SETQ IllegalCopyFlg T))
([AND (EQ Label 'SubBox)
      (OR (EQ NewSourceCard DestinationCard)
          (NOT (NC.NotDaughterP DestinationCard NewSourceCard (FUNCTION NC.ChildLinkP)
                               ; Copy to a filebox causes a cycle.
(NC.PrintMsg Window NIL (CONCAT "Couldn't copy " Link " because of subbox cycle." (CHARACTER 13)))
(SETQ IllegalCopyFlg T))
((AND (FMEMB Label NC.SystemLinkLabels)
      [NOT (FMEMB Label '(FiledCard SubBox))
          (NEQ NewSourceCard SourceCard))
      ; Copy of system link outside of own card.
(NC.PrintMsg Window NIL (CONCAT "Tried to copy system link." (CHARACTER 13)
                                "Link type of copy set to 'Unspecified'."
                                (CHARACTER 13)))
(SETQ NewLabel NC.UnspecifiedLinkLabel))
((NC.CrossFileLinkCardP DestinationCard)
      ; Copy of a cross-file link.
(if (NOT (NC.ActiveCardP DestinationCard))
    then (NC.GetNoteCard DestinationCard)
      ; If it's to be two-way, then make a fresh CrossFileLinkCard.
(if (OR (EQ NC.CopyCrossFileLinksMode 'TWOWAY)
        (AND (NULL NC.CopyCrossFileLinksMode)
             NC.NewCrossFileLinksTwoWayFlg))
    then (SETQ DestinationCard (NC.GetCrossFileLinkDestCard DestinationCard Window)))
(OR DestinationCard (SETQ IllegalCopyFlg T))
[COND
  (IllegalCopyFlg (NC.ReplaceWithDeletedLinkImageObj ImageObjectCopy)
                  (NC.SetUserDataProp SourceCard 'NeedsCleaningFlg T))
  (ExternalCopyFlg (NC.CoerceToExternalPutLinkIcon ImageObjectCopy SourceCard DestinationCard
                                                    (OR NewLabel Label)))
  (T (NC.SetUserDataProp NewSourceCard 'LastLinkCopiedOrMoved (NC.FillInLinkIcon ImageObjectCopy
                                          (OR NewLabel Label)
                                          DestinationCard NewSourceCard
                                          OldDisplayMode (NC.LinkAtCharPos
                                                           InsertPos
                                                           ToTextStream)
                                          NC.CopyCrossFileLinksMode]
(IMAGEOBJPROP ImageObject 'ImageObjectCopy NIL))

```

**(NC.LinkIconWhenDeletedFn**

[LAMBDA (ImageObject Stream)

; Edited 9-Dec-88 18:58 by krivacic

;;; bk 12/9/88 Add the OriginalImageObject property to the resulting IMAGEOBJ

;;; When deleting a link icon image object from a card, make sure the link information on the source and destination cards get updated. When deleting a link image object from a prop list editor, just inform the user that this is a useless operation.

;;; rht 11/19/84: Now handles the case when a move is in progress. True if the old link is cached on the ImageObj. Also changed so that SourceID is computed from ImageObject rather than from Stream. Similarly for the window.

;;; kirk 14Nov85: deleted use of PSA.Database

;;; fgh 5/2/86 Added code to handle deleting of links from ShowLinks editors.

;;; fgh 6/4/86 Added KLUDGE to take care of case when this is called under TEDIT.UNDO.MOVE

;;; rht 7/4/86: Now checks for readonly cards. Also passes checks for deleting from orphan card when calling NC.DeleteLink.

;;; fgh 7/14/86 Now deletes link icon directly, rather than depending on DeleteLink to do it. Sets the Don'tDeleteLinkIconFlg in the call to DeleteLink to do so.

;;; rht 8/11/86: Now computes InsideShowLinksP and lets that determine value of Don'tDelLinkIcon flg in call to NC.DeleteLink.

;;; rht&pmi 12/8/86: Now checks whether we're called by TEDIT.MOVE. If so, don't cause orphan hook under NC.DelFromLink.

;;; rht 3/26/87: Now passes non-nil NoOrphanHookFlg to NC.DeleteLink in case we're deleting a cross-file link.

;;; rht 10/1/87: Fixed case when we're under a move to call NC.DeleteLink rather than NC.DelFromLink and NC.DelToLink individually.

;;; rht 12/9/87: Now checks to make sure links are cached before calling NC.CheckOrphanDelete.

```
(DECLARE (GLOBALVARS NC.DeletedLinkImageObject))
(LET ((Link (NC.FetchLinkFromLinkIcon ImageObject))
      LinkType OldLink StkPtr SourceCard DestinationCard InsideShowLinksP)
  (COND
    [(NC.CheckForNotReadOnly (fetch (Link SourceCard) of Link)
      NIL "Can't delete links in ")
      (COND
        ((IMAGEOBJPROP ImageObject 'LinkBeingMovedWithinCard)
          ; TEdit is trying to delete the old link imageobj after moving within
          ; a card.
          (IMAGEOBJPROP ImageObject 'LinkBeingMovedWithinCard NIL))
        ((SETQ OldLink (IMAGEOBJPROP ImageObject 'LinkBeingMoved))
          (NC.DeleteLink OldLink)
          (replace (Link UID) of OldLink with -1)
          (IMAGEOBJPROP ImageObject 'LinkBeingMoved NIL))
        ((SETQ StkPtr (STKPOS 'TEDIT.UNDO.MOVE))
```

;;; Called from TEDIT.UNDO.MOVE, don't rely want to delete the links. KLUDGE because TEDIT will not call the LinkIconInsertFn later on.

```
(RELSTK StkPtr)
NIL)
(T
```

;;; if this is reversed source/dest (i.e., a From link) inside the show links editor, then return it to its original direction

```
[COND
  ((AND (SETQ InsideShowLinksP (IMAGEOBJPROP ImageObject 'InsidePropListEditor))
        (LISTGET (fetch (Link UserData) of Link)
          'Reversed))
    (SETQ Link (create Link using Link SourceCard _ (fetch (Link DestinationCard) of Link)
      DestinationCard _ (fetch (Link SourceCard) of Link)
```

;;; Then just delete the link

```
(SETQ SourceCard (fetch (Link SourceCard) of Link))
(SETQ DestinationCard (fetch (Link DestinationCard) of Link))
(LET ((WasCachedFlg (NC.LinksCachedP SourceCard))
      (if (NOT WasCachedFlg)
        then (NC.GetLinks SourceCard))
      (NC.CheckForOrphanDelete SourceCard DestinationCard)
      (NC.DeleteLink Link (OR (NC.SameCardP SourceCard (fetch (NoteFile OrphansCard)
        of (fetch (Card NoteFile) of SourceCard)
      ))
      (NC.CrossFileLinkCardP DestinationCard)
      (NC.CrossFileLinkCardP SourceCard))
      (NOT InsideShowLinksP))
      (if (NOT WasCachedFlg)
        then (NC.PutLinks SourceCard)
        (NC.UncacheLinks SourceCard)))
```

;; Add the OriginalImageObject property to the new IMAGEOBJ, so the LINK will come back in a TEdit UNDO

```
(create IMAGEOBJ
  OBJECTDATUM _ (fetch (IMAGEOBJ OBJECTDATUM) of NC.DeletedLinkImageObject)
  IMAGEOBJPLIST _ (CONS 'OriginalImageObject (CONS ImageObject (fetch (IMAGEOBJ
    IMAGEOBJPLIST
  )
  of
  NC.DeletedLinkImageObject
  )))
  IMAGEOBJFNS _ (fetch (IMAGEOBJ IMAGEOBJFNS) of NC.DeletedLinkImageObject))
(replace (Link UID) of Link with -1]
(T (ERROR!))
```

**(NC.LinkIconWhenInsertedFn**

[LAMBDA (ImageObj WindowStream)

; Edited 9-Dec-88 18:56 by krivacic

;;; Make a new UID for a LINK which is being re-inserted (i.e. it's UID is -1)

;;; An Notecard link is being inserted. If this is a deleteundo, then have to re-establish the link information



```

(NC.PrintMsg Window NIL (CONCAT "Tried to move a NoteCards link icon" " to a non-NoteCards
stream!!." (CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
((IMAGEOBJPROP ImageObject 'InsidePropListEditor) ; In a ShowLinks window
(NC.PrintMsg Window NIL (CONCAT "Cannot move from a Show Links Window." (CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
([NULL (NC.LinksSupportedP NewSourceCard '(Local)
(NC.PrintMsg Window NIL (CONCAT "Tried to move a NoteCards link icon" " to a NoteCard that" " does
not support links!!." (CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
(NC.SameCardP NewSourceCard DestinationCard)
(NC.PrintMsg Window NIL (CONCAT "Sorry, links aren't allowed to point from cards to themselves."
(CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
((AND (FMEMB Label '(FiledCard SubBox))
(NEQ (NC.RetrieveType NewSourceCard
'FileBox)) ; Move from filebox to non-filebox.
(NC.PrintMsg Window NIL (CONCAT "Tried to move filedcard or subbox link to a non-filebox."
(CHARACTER 13)
"Link type of copy set to 'Unspecified'."
(CHARACTER 13)))
(SETQ NewLabel NC.UnspecifiedLinkLabel))
((AND (NEQ (NC.RetrieveType SourceCard
'FileBox)
(EQ (NC.RetrieveType NewSourceCard
'FileBox)) ; Move from non-filebox to filebox.
(NC.PrintMsg Window NIL (CONCAT "Can't move links from non-filebox to filebox." (CHARACTER 13)
"Try using PutCardsHere."
(CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
([AND (EQ (NC.RetrieveType NewSourceCard
'FileBox)
(NEQ NewSourceCard SourceCard)
(for Link1 in (NC.FetchToLinks NewSourceCard) thereis (AND (NC.ChildLinkP Link1)
(EQ DestinationCard
(fetch (Link DestinationCard)
of Link1]
; Move to a filebox already containing this child.
(NC.PrintMsg Window NIL (CONCAT (NC.RetrieveTitle DestinationCard)
" not moved: already appears as a child of "
(NC.RetrieveTitle NewSourceCard)
(CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
([AND (EQ Label 'SubBox)
(NEQ NewSourceCard SourceCard)
(OR (EQ NewSourceCard DestinationCard)
(NOT (NC.NotDaughterP DestinationCard NewSourceCard (FUNCTION NC.ChildLinkP]
; Move to a filebox causes a cycle.
(NC.PrintMsg Window NIL (CONCAT "Couldn't move " Link " because of subbox cycle." (CHARACTER 13)))
(RETFROM 'TEDIT.MOVE NIL T))
((AND (FMEMB Label NC.SystemLinkLabels)
[NOT (FMEMB Label '(FiledCard SubBox)
(NEQ NewSourceCard SourceCard)) ; Move of system link outside of own card.
(NC.PrintMsg Window NIL (CONCAT "Tried to copy system link." (CHARACTER 13)
"Link type of copy set to 'Unspecified'."
(CHARACTER 13)))
(SETQ NewLabel NC.UnspecifiedLinkLabel))
((AND (NC.CrossFileLinkCardP DestinationCard)
(NOT (NC.GetCrossFileLinkDestCard DestinationCard Window)))
; Trying to move a cross-file link, but can't get to it so bail out.
(RETFROM 'TEDIT.MOVE NIL T))
(if (EQ SourceCard NewSourceCard)
then ; Moving within a card, so just reposition Link in ToLinks list.
(AND (NC.TeditBasedP SourceCard)
(IMAGEOBJPROP ImageObject 'LinkBeingMovedWithinCard Link))
(PROG1 (NC.AddToLink Link (NC.LinkAtCharPos InsertPos ToTextStream)
T)
(NC.SetUserDataProp SourceCard 'LastLinkCopiedOrMoved Link))
else (IMAGEOBJPROP ImageObject 'LinkBeingMoved Link)
(NC.SetUserDataProp NewSourceCard 'LastLinkCopiedOrMoved (NC.FillInLinkIcon ImageObject
(OR NewLabel Label)
DestinationCard NewSourceCard
OldDisplayMode (NC.LinkAtCharPos
InsertPos
ToTextStream]))

```

(NC.FetchLinkFromLinkIcon

[LAMBDA (LinkIcon)

; Edited 9-Dec-88 14:21 by krivacic

;;; Return the link information from the LinkIcon image object

;;; rht 7/18/85: Changed to convert to typed links.

;;; fgh 11/16/85 Removed code to make link into aTYPEDRECORD, since links are now datatypes.

(NC.CheckDisplayModeFormat (IMAGEOBJPROP LinkIcon 'OBJECTDATUM])

**(NC.ChangeLinkDisplayMode**

[LAMBDA (LinkOrLinkIcon Window NewDisplayMode)

; Edited 9-Jan-89 10:20 by rtk

;;; Change the display mode of the NOTECARDLINK defined by LinkOrLinkIcon. Have the user choose the new display mode.

;;; NOTE: assumes that the SourceID card of the link is active, i.e., the substance and links are cached for this card.

;;; rht 2/1/85: Now doesn't do all the work if new display mode is same as the old one. Or if user doesn't make a selection from the menu.

;;; kef 8/8/86: Now does the write through to the NoteFile even if the card is active, but not being edited. Also, in the case of doing the write through, puts only the from links.

;;; fgh 8/30/86 Adapted to call NC.IfCardPartNotBusy.

;;; t 9/29/86: Changed Ken's call to NC.CardBeingEditedP to NC.ActiveCardP and other minor changes.

;;; rht 2/18/87: Added check for readonly notefile to fix bug #376.0

;;; rg 5/15/87 now calls NC.LinksCachedP instead of NC.ActiveCardP when changing FromLinks

;;; bk 1/9/89 Redisplay the link icons even for tedit windows.

```
(LET (Card DestinationCard Link LinkIcon GlobalLinkFlg OldDisplayMode)
  [if (NC.LinkIconImageObjP LinkOrLinkIcon)
    then (SETQ Link (NC.FetchLinkFromLinkIcon LinkOrLinkIcon))
        (SETQ LinkIcon LinkOrLinkIcon)
    else (SETQ Link LinkOrLinkIcon)
        (OR (SETQ GlobalLinkFlg (NC.GlobalLinkP Link))
            (SETQ LinkIcon (NC.FetchLinkIconForLink Link))
            (SETQ Card (fetch (Link SourceCard) of Link))
            (SETQ OldDisplayMode (fetch (Link DisplayMode) of Link))
            (if (NC.CheckForNotReadOnly Card Window "Can't change display mode for links in ")
                then [COND
                    ((WINDOWP Window))
                    ((NC.ActiveCardP Card)
                     (SETQ Window (NC.FetchWindow Card))

```

;;; Get new display mode from user

```
(SETQ DestinationCard (fetch (Link DestinationCard) of Link))
(NC.IfCardPartNotBusy DestinationCard 'FROMLINKS
 [if (NOT NewDisplayMode)
   then (SETQ NewDisplayMode
          (NC.DisplayModeFromStylesheetSelections
           (STYLESHEET (CREATE.STYLE 'ITEMS [LIST (create MENU
                                                    ITEMS _ '(Yes No
                                                         System% Defaults)
                                                    )
              (create MENU
                ITEMS _ '(Yes No
                         System% Defaults)
              )
              (create MENU
                ITEMS _ '(Yes No
                         System% Defaults)
              )
              'ITEM.TITLES
              '(Title?% LinkType?% AttachBitmap?)
              'ITEM.TITLE.FONT
              (FONTCOPY MENUFONT 'WEIGHT 'BOLD)
              'SELECTIONS
              (NC.StylesheetSelectionsFromDisplayMode OldDisplayMode)
              'NEED.NOT.FILL.IN NIL 'TITLE 'Display% Mode? 'POSITION
              (create POSITION
                XCOORD _ LASTMOUSEX
                YCOORD _ LASTMOUSEY]
           )
   (if (OR (NULL NewDisplayMode)
           (EQUAL NewDisplayMode OldDisplayMode))
     then

```

;;; If new display mode is same as old, then bail out.

```
      NIL
    else

```

;;; Update link in LinkIcon or in global links list

```
(if GlobalLinkFlg
  then (for GlobalLink in (NC.FetchGlobalLinks Card)
        when (NC.SameLinkP Link GlobalLink)
        do (replace (Link DisplayMode) of GlobalLink with NewDisplayMode))
      (NC.SetLinksDirtyFlg Card T)
  else (replace (Link DisplayMode) of (NC.FetchLinkFromLinkIcon LinkIcon) with
              NewDisplayMode
             )
      (NC.MarkCardDirty Card))

```

::: Update link in ToLinks List of Source card

```
(for ToLink in (NC.FetchToLinks Card) when (NC.SameLinkP Link ToLink)
do (replace (Link DisplayMode) of ToLink with NewDisplayMode))
(NC.SetLinksDirtyFlg Card T)
```

::: Update Link in FromLinks list of Destination card

```
(if (NC.LinksCachedP DestinationCard)
then (for FromLink in (NC.FetchFromLinks DestinationCard)
when (NC.SameLinkP Link FromLink)
do (replace (Link DisplayMode) of FromLink with NewDisplayMode))
(NC.SetLinksDirtyFlg DestinationCard T)
else (WITH.MONITOR (NC.FetchMonitor (fetch (Card NoteFile) of DestinationCard))
(NC.GetLinks DestinationCard)
(for FromLink in (NC.FetchFromLinks DestinationCard)
when (NC.SameLinkP Link FromLink)
do (replace (Link DisplayMode) of FromLink with NewDisplayMode))
(NC.PutFromLinks DestinationCard)
(NC.UncacheLinks DestinationCard)))
```

::: Update the image in any active window for Card

```
(if (WINDOWP Window)
then ;: Update the Object in the window (do this for ALL windows now since we are a seperate process.
(NC.UpdateLinkImages Window (fetch (Link DestinationCard) of Link)))
; This used to cause tedit to redisplay the screen.
' CHANGED])
```

### (NC.UpdateLinkImages

[LAMBDA (SourceWindowOrCard DestinationCard) (\* rht%: "30-Sep-87 19:35")

```
(* * Update the Link Image Objects in SourceCard that point to DestinationID)
(* * rht 10/2/85%: Now checks if card's window was shrunk. If so, unshrink, modify and reshink.)
(* * fgh |11/17/85| Updated to handle card objects.)
(* * fgh |2/5/86| Addeed call to NC.ApplyFn)
(* * rht 9/30/87%: Added calls to NC.UpdateLinkImagesInShowLinks.)
```

```
(LET (SourceWindow SourceCard NoteCardType ShrunkWin)
(COND
((WINDOWP SourceWindowOrCard)
(SETQ SourceWindow SourceWindowOrCard)
(SETQ SourceCard (NC.CoerceToCard SourceWindowOrCard)))
(T (SETQ SourceWindow (NC.FetchWindow SourceWindowOrCard))
(SETQ SourceCard SourceWindowOrCard)))
(COND
((SETQ ShrunkWin (NC.GetShrunkWin SourceWindow))
(EXPANDW ShrunkWin)))
(SETQ NoteCardType (NC.RetrieveType SourceCard))
(NC.UpdateLinkImagesInShowLinks SourceCard DestinationCard)
(NC.UpdateLinkImagesInShowLinks DestinationCard SourceCard)
(NC.ApplyFn UpdateLinkIconsFn SourceCard DestinationCard)
(COND
(ShrunkWin (SHRINKW SourceWindow)))
NIL])
```

### (NC.LinkIconCopyFn

[LAMBDA (ImageObject) ; Edited 5-Jan-89 17:14 by rtk

::: Return a copy of image object. IF it is a system link label, then make its label unspecified.

::: rht 11/18/84: No longer worries about system link labels. All that is handled by WhenCopiedFn and WhenMovedFn. Caches the new copy on the  
::: prop list of the original imageobj.

::: rht 8/22/85: Added call to NC.CoerceToExternalPutLinkIcon to make the imageobj look like a external link icon. If we're in a normal imageobj copy  
::: then the whencopiedfn will get called and replace the contents of ImageObjectCopy anyway. However, if only the CopyFn gets called, then this will  
::: stick.

::: fgh 1/29/86 Removed the 'copying ImageObj' clause in the create of ImageObjectCopy to fix bug where the copying was causing a stack overflow thru  
::: infinite recursive COPYALL calls. Copying clause was unnecessary.

::: rht 6/24/87: Now saves original imageobj on the OriginalImageObject property of ImageObjectCopy.

::: bk 1/5/89 Set the ImageObjectCopy AND OriginalImageObject Properties before callingNC.CoerceToExternalPutLinkIcon. Also Make the Image  
::: Object

```
(LET ((ImageObjectCopy (BITMAPEDITOBJ))
(Link (NC.FetchLinkFromLinkIcon ImageObject)))
(IMAGEOBJPROP ImageObject 'ImageObjectCopy ImageObjectCopy)
```

```
(IMAGEOBJPROP ImageObjectCopy 'OriginalImageObject ImageObject)
(NC.CoerceToExternalPutLinkIcon ImageObjectCopy (fetch (Link SourceCard) of Link)
 (fetch (Link DestinationCard) of Link)
 (fetch (Link Label) of Link))
(IMAGEOBJPROP ImageObjectCopy 'WHENINSERTEDFN (FUNCTION NC.LinkIconWhenInsertedFn)
ImageObjectCopy])
```

**(NC.LinkAtCharPos**

```
[LAMBDA (CharPos TextStream) (* rht%: " 1-Feb-87 15:45")
```

(\* Returns the link right before CharPos in TextStream. Return NIL if there is no link icon located before CharPos.)

(\* rht 1/25/87%: Now takes special kludgy action if we're under a copy of multiple links. If it's past the first link in the copy then the value returned is the link just copied previously.)

(\* rht 2/1/87%: Fix to above kludgy action so that STKARG call is safe.)

```
(if (TEXTSTREAMP TextStream)
  then (if [LET [(StkPos (STKPOS 'TEDIT.SELECTED.PIECES)
 (AND StkPos (PROG1 (STKARG 'NPC StkPos)
 (RELSTK StkPos)
 then (NC.FetchUserDataProp (NC.CoerceToCard TextStream)
 'LastLinkCopiedOrMoved)
 else (for LinkIconPair in (NC.TEditCollectObjects TextStream (FUNCTION NC.LinkIconImageObjP))
 bind OldLinkIconPair while (LESSP (CADR LinkIconPair)
 CharPos)
 do (SETQ OldLinkIconPair LinkIconPair) finally (RETURN (AND OldLinkIconPair
 (NC.FetchLinkFromLinkIcon
 (CAR OldLinkIconPair))
```

**(NC.LinkIconLeftButtonFn**

```
[LAMBDA (LinkIcon Window) ; Edited 30-May-88 15:23 by Trigg
```

;;; Left button has been pressed in the link icon. For now, just traverse the link.

;; rht 5/30/88: Now checks to see if destination card type has a LinkIconLeftButtonFn prop. If so, run that instead of traversing the link.

```
(LET ((Card (fetch (Link DestinationCard) of (NC.FetchLinkFromLinkIcon LinkIcon)))
 CardType LinkIconLeftButtonFn)
 (SETQ CardType (NC.RetrieveType Card))
 (if (SETQ LinkIconLeftButtonFn (GETPROP CardType 'LinkIconLeftButtonFn))
 then (ADD.PROCESS '(APPLY* ',LinkIconLeftButtonFn ',Card ',Window)
 'NAME
 'LinkIconLeftButton)
 else (NC.TraverseLink LinkIcon Window]))
```

**(NC.LinkToBitMap**

```
[LAMBDA (IMAGE-OBJ) ; Edited 6-Jan-89 16:19 by rtk
```

;;; Return the BitMap for the Link Image object.

```
(DSPFONT NC.LinkIconFont NC.LinkIconBitMapStream)
(DSPYPOSITION 0 NC.LinkIconBitMapStream)
(DSPYPOSITION 0 NC.LinkIconBitMapStream)
(LET* ((LINK-BOX (OR (IMAGEOBJPROP IMAGE-OBJ 'BOUNDBOX)
 (NC.LinkIconImageBoxFn IMAGE-OBJ NC.LinkIconBitMapStream)))
 (DEST-BITMAP (BITMAPCREATE (fetch (IMAGEBOX XSIZE) of LINK-BOX)
 (fetch (IMAGEBOX YSIZE) of LINK-BOX)))
 (BITMAP-STREAM (DSPCREATE DEST-BITMAP)))
 ;; Set the Position above the descent
 (DSPYPOSITION (fetch (IMAGEBOX YDESC) of LINK-BOX)
 BITMAP-STREAM)
 ;; Draw the bitmap
 (NC.LinkIconDisplayFn IMAGE-OBJ BITMAP-STREAM)
 ;; return the resulting bitmap
 (CL:VALUES DEST-BITMAP (OR (IMAGEOBJPROP IMAGE-OBJ 'BOUNDBOX)
 LINK-BOX]))
```

**(NC.SetBitMap&Box**

```
[LAMBDA (IMAGE-OBJ BITMAP) ; Edited 6-Jan-89 16:19 by rtk
```

;;; Set the IMAGEBOX and IMAGEBITMAP properties of IMAGE-OBJ. Return both BITMAP & IMAGEBOX.

```
(LET (DEST-IMAGEBOX)
 [if (NULL BITMAP)
 then (CL:MULTIPLE-VALUE-BIND (NEW-BITMAP NEW-IMAGEBOX)
 (NC.LinkToBitMap IMAGE-OBJ)
 (SETQ BITMAP NEW-BITMAP)
 (SETQ DEST-IMAGEBOX NEW-IMAGEBOX)
 (IMAGEOBJPROP IMAGE-OBJ 'IMAGEBITMAP BITMAP))
 else (SETQ DEST-IMAGEBOX (OR (IMAGEOBJPROP IMAGE-OBJ 'BOUNDBOX)
```

```

        (create IMAGEBOX
          XSIZE _ (BITMAPWIDTH BITMAP)
          YSIZE _ (BITMAPHEIGHT BITMAP)
          YDESC _ 0
          XKERN _ 0]
      (IMAGEOBJPROP IMAGE-OBJ 'IMAGEBOX DEST-IMAGEBOX)
      (CL:VALUES BITMAP DEST-IMAGEBOX])
)
(DEFINEQ
(NC.FillInLinkIcon
  [LAMBDA (ImageObj Label DestinationCard SourceCard DisplayMode LinkToInsertAfter CrossFileLinksMode)
    (* rht%: "25-Jan-87 20:01")

    (** Called by WhenCopiedFn and WhenMovedFn to build the new link and stick it in the ImageObj.)

    (** rht 1/25/87%: Now returns the new link.)

    (DECLARE (GLOBALVARS NC.LinkIconImageFns))
    (LET ((Link (NC.MakeLink NIL Label DestinationCard SourceCard DisplayMode NIL NIL NIL LinkToInsertAfter
      CrossFileLinksMode)))
      (IMAGEOBJPROP ImageObj 'OBJECTDATUM Link)
      (replace (IMAGEOBJ IMAGEOBJFNS) of ImageObj with NC.LinkIconImageFns
        Link])
)

```

```

(NC.InvisibleLinkImageBoxFn
  [LAMBDA (ImageObj Stream)
    (* rht%: "20-Nov-84 10:57")

    (** Returns a zero-size image box.)

    (create IMAGEBOX
      XSIZE _ 0
      YSIZE _ 0
      YDESC _ 0
      XKERN _ 0])
)

```

```

(NC.InvisibleLinkPutFn
  [LAMBDA (ImageObj OutputStream)
    (BOUT OutputStream 1])
  (* fgh%: "26-Feb-84 01:04")
)

```

```

(NC.InvisibleLinkGetFn
  [LAMBDA (InputStream TextStream)
    (BIN InputStream)
    NC.InvisibleLinkImageObject])
  (* rht%: "21-Nov-84 12:10")
)

```

;;; Pointer icons. These act like links, except that their icon boxes are dashed, and they're invisible as far as the source or destination cards are concerned. That is, only the imageobject containing the pointer knows about the connection. Functions exist for creating pointer icons and converting them into link icons.

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.PointerIconImageObjFns NC.PointerIconDashingStyle NC.PointerIconMiddleButtonMenu)
)

(RPAQ? NC.PointerIconImageObjFns (IMAGEFNSCREATE (FUNCTION NC.PointerIconDisplayFn)
  (FUNCTION NC.PointerIconImageBoxFn)
  (FUNCTION NC.PointerIconPutFn)
  (FUNCTION NC.PointerIconGetFn)
  (FUNCTION NC.PointerIconCopyFn)
  (FUNCTION NC.LinkIconButtonEventInFn)))
)

```

```
(RPAQ? NC.PointerIconDashingStyle ' (4 2))
```

;;; Functions likely to be useful externally.

```

(DEFINEQ
(NC.InsertPointerInText
  [LAMBDA (SourceCard DestinationCard PointerOrPointerLabel DisplayMode CharacterPosition)
    (* rht%: "27-Oct-87 12:23")

    (** Create a Pointer icon pointing to DestinationCard and stick it in SourceCard at location Loc.
      Loc defaults to current cursor position. Return the Pointer icon.)

    (LET ((WasActiveFlg (NCP.CardCachedP SourceCard))
      Pointer TextStream InsertPos)
      (if (NOT WasActiveFlg)
        then (NCP.CacheCards SourceCard))
)
)

```







```

Card ShowTitleFlg AttachBitmapFlg ShowLinkTypeFlg)
;; Save CurrentX and RightMargin on ImageObj
  (IMAGEOBJPROP ImageObj 'CurrentX CurrentX)
  (IMAGEOBJPROP ImageObj 'RightMargin RightMargin)
;; Check for bad Pointer
  (if [NOT (NC.ValidCardP (SETQ Card (fetch (NCPPointer DestinationCard) of Pointer)]
    then ;; Delete the offending link, smash its image object into a 'Deleted' image object, and return by displaying it.
      (RETURN (NC.DeletedLinkImageBoxFn ImageObj ImageStream)))
;; Determine what type of Display to do
  (OR DisplayType (SETQ DisplayType (fetch (NCPPointer DisplayMode) of Pointer)))
  (SETQ ShowTitleFlg (fetch (LINKDISPLAYMODE SHOWTITLEFLG) of DisplayType))
  (SETQ ShowLinkTypeFlg (fetch (LINKDISPLAYMODE SHOWLINKTYPEFLG) of DisplayType))
  (SETQ AttachBitmapFlg (fetch (LINKDISPLAYMODE ATTACHBITMAPFLG) of DisplayType))
  (SETQ AttachBitmapFlg (if (EQ AttachBitmapFlg 'FLOAT)
    then NC.LinkIconAttachBitmapFlg
    else AttachBitmapFlg))
;; Construct the text for the link icon
  (if [NOT LinkIconString]
    then (OR Title (SETQ Title (if (AND ShowTitleFlg (OR (NEQ ShowTitleFlg 'FLOAT)
      NC.LinkIconShowTitleFlg))
      then (NC.RetrieveTitle Card)
      else NIL)))
    [OR Label (SETQ Label (AND (if (EQ ShowLinkTypeFlg 'FLOAT)
      then NC.LinkIconShowLinkTypeFlg
      else ShowLinkTypeFlg)
      (fetch (NCPPointer Label) of Pointer)]
    (SETQ LinkIconString (CONCAT (COND
      (Label (CONCAT "<" Label ">"))
      (T ""))
      (COND
        ((AND Label Title)
         " ")
        (T ""))
      (OR Title ""))
;; Now do the hard work.
  (RETURN (NC.ComputeLinkOrPointerImageBox ImageStream LinkIconString AttachBitmapFlg Card RightMargin
    (IMAGEOBJPROP ImageObj 'BorderWidthIncrement)
    CurrentX))))]
```

**(NC.PointerIconPutFn**

[LAMBDA (ImageObject Stream) ; Edited 21-Jun-88 09:21 by pmi

```

;; Writes the Pointer information in the pointer icon specified by ImageObject onto Stream
;; pmi 6/21/88: Now checks that pointer has a valid destination card before trying to write it out. If it doesn't, convert it to a deleted imaged obj.
(LET ((Pointer (NC.FetchLinkFromLinkIcon ImageObject)))
  (if (type? NCPPointer Pointer)
    then ;; Check for bad Pointer
      (if (NOT (NC.ValidCardP (fetch (NCPPointer DestinationCard) of Pointer)))
        then ;; Delete a bad pointer when it is discovered at put time.
          (NC.ReplaceWithDeletedLinkImageObj ImageObject)
          (APPLY* (IMAGEOBJPROP ImageObject 'PUTFN)
            ImageObject Stream)
        else (NC.WritePointer Pointer Stream))])
```

**(NC.PointerIconGetFn**

[LAMBDA (Stream) ; Edited 22-Jun-88 12:37 by pmi

```

;; Reads the pointer information from Stream and returns a pointer icon image object.
;; pmi 6/22/88: Now checks that NC.ReadPointer returns a valid NCPPointer object before calling NC.MakePointerIcon. If not, returns
;; NC.DeletedLinkImageObject
(DECLARE (GLOBALVARS NC.DeletedLinkImageObject))
(LET ((Pointer (NC.ReadPointer Stream)))
  (if (type? NCPPointer Pointer)
    then (NC.MakePointerIcon Pointer)
    else NC.DeletedLinkImageObject])
```

**(NC.PointerIconCopyFn**

[LAMBDA (PointerIcon) (\* rht%: "29-Oct-87 12:33")

(\* \* Return a copy of this pointer icon.)

```

(LET ((Pointer (NC.FetchPointerFromPointerIcon PointerIcon)))
  (NC.MakePointerIcon (NC.MakePointer (fetch (NCPPointer DestinationCard) of Pointer)
    (fetch (NCPPointer Label) of Pointer)
    (fetch (NCPPointer DisplayMode) of Pointer)]))
```

**(NC.WritePointer**

[LAMBDA (Pointer Stream)

(\* rht%: "24-Oct-87 23:05")

(\* \* Write a single Pointer DATATYPE instance down to Stream)

```
(DECLARE (GLOBALVARS NC.OrigReadTable))
(LET ((DestinationCard (fetch (NCPPointer DestinationCard) of Pointer)))
  (* * This is version 0 Pointer format)

  (NC.WritePtr Stream 0 1)
  (NC.WriteUID Stream (fetch (Card UID) of DestinationCard))
  (NC.WriteUID Stream (fetch (NoteFile UID) of (NCP.CardNoteFile DestinationCard)))
  (PRINT (fetch (NCPPointer Label) of Pointer)
    Stream NC.OrigReadTable)
  (PRINT (fetch (NCPPointer DisplayMode) of Pointer)
    Stream NC.OrigReadTable])
```

**(NC.ReadPointer**

[LAMBDA (Stream)

; Edited 22-Jun-88 12:11 by pmi

:: Read a single Pointer DATATYPE instance from Stream

:: pmi 6/22/88: Now checks for a 1 as the first character in the stream, signifying a Deleted link image obj.

```
(DECLARE (GLOBALVARS NC.OrigReadTable NC.DeletedLinkImageObject))
(LET ((FirstChar (NC.ReadPtr Stream 1)))
  (COND
    ((EQ FirstChar 0)
     ;; The link info for a version 0 style pointer.
     (PROG1 (create NCPPointer
       DestinationCard _ (NC.CardOrCardHolderFromUID (NC.ReadUID Stream)
         (NC.ReadUID Stream))
       Label _ (READ Stream NC.OrigReadTable)
       DisplayMode _ (READ Stream NC.OrigReadTable))
```

::: Get that damn CR

```
(BIN Stream)))
((EQ FirstChar 1)
 NC.DeletedLinkImageObject)
((EQ FirstChar 13)
```

::: KLUDGE to account for fact that some versions of WRITE.IMAGEOBJ put a CR before beginning the actual stuff.

(NC.ReadPointer Stream])

)

(DEFINEQ

**(NC.PointerIconLeftButtonFn**

[LAMBDA (PointerIcon Window)

(\* rht%: "25-Oct-87 00:42")

(\* \* Left button has been pressed in the pointer icon. For now, just traverse the pointer.)

(NC.TraversePointer PointerIcon Window])

**(NC.PointerIconMiddleButtonFn**

[LAMBDA (PointerIcon Window)

(\* rht%: "29-Oct-87 12:09")

(\* \* Middle button has been pressed in a pointer icon. Bring up menu of operations.)

```
(DECLARE (GLOBALVARS NC.PointerIconMiddleButtonMenu NC.MenuFont))
(LET ((Card (fetch (NCPPointer DestinationCard) of (NC.FetchPointerFromPointerIcon PointerIcon)))
  ExtraItems Selection Menu)
  (SETQ ExtraItems (GETPROP (NC.RetrieveType Card)
    'ExtraLinkIconMenuItems))
  [SETQ Menu (COND
    ((AND (BOUNDP 'NC.PointerIconMiddleButtonMenu)
      (type? MENU NC.PointerIconMiddleButtonMenu)
      (NULL ExtraItems))
     NC.PointerIconMiddleButtonMenu)
    (T (create MENU
      ITEMS _ (APPEND '( |Bring Up Card/Box| 'Edit "Bring up the note card pointed to
        by this pointer."
        | |Change Link Type| 'Relabel "Change the label of this
        pointer."
        | |Change Card Title| 'Retitle "Change the destination card's
        title."
        | |Change Display Mode| 'Display "Change the type of
        information displayed in this pointer."
        | |Convert to Link| 'Convert "Turn this pointer icon into a
        link icon."))
```

```

                                ExtraItems)
                                CENTERFLG _ T
                                TITLE _ "Link Ops"
                                MENUFONT _ NC.MenuFont
                                ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                                1])
(if (NULL ExtraItems)
  then (SETQ NC.PointerIconMiddleButtonMenu Menu))
(if (NC.CardP (NC.CoerceToCard Window))
  then (SELECTQ (SETQ Selection (MENU Menu))
            (Edit (NC.TraversePointer PointerIcon Window))
            (Display (NC.ChangePointerDisplayMode PointerIcon Window))
            (Relabel (NC.RelabelPointer PointerIcon Window))
            (Retitle (NC.ChangeCardTitleFromPointerIcon PointerIcon Window))
            (Convert (NC.ConvertPointerIconToLinkIcon Window PointerIcon)
                    'CHANGED)
            (NIL NIL)
            (APPLY* Selection Card))
  else NIL))

```

**(NC.TraversePointer**

```
[LAMBDA (PointerIcon Window) (* rht%: "25-Oct-87 00:45")
```

(\* Follow the pointer to the card at the other end. For now, amounts to editing the destination card.)

```

(DECLARE (GLOBALVARS NC.MsgDelay))
(LET ((Pointer (NC.FetchPointerFromPointerIcon PointerIcon))
      (InterestedWindow (NC.CoerceToInterestedWindow (MAINWINDOW Window)))
      (Card OldTitle)
      (SETQ Card (fetch (NCPointer DestinationCard) of Pointer)))
  (if (NC.ValidCardP Card)
    then (SETQ OldTitle (NC.RetrieveTitle Card))
        (NC.EditNoteCard Card))

```

(\* We're under TEdit, so if title was changed by EditFn, then be sure link icon gets redisplayed.)

```

  (if (STREQUAL OldTitle (NC.RetrieveTitle Card))
    then 'DON'T
    else 'CHANGED)

```

else

(\* The pointer is bad - tell the user and change it to "Deleted")

```

  (NC.PrintMsg InterestedWindow T "This is a bad pointer - the destination card is invalid."
    (CHARACTER 13))
  (DISMISS NC.MsgDelay)
  (NC.ClearMsg InterestedWindow T)
  (NC.DeleteBadPointer Pointer PointerIcon Window])

```

**(NC.ChangePointerDisplayMode**

```
[LAMBDA (PointerIcon Window NewDisplayMode) (* rht%: "25-Oct-87 01:00")
```

(\* Change the display mode of the pointer icon defined by LinkOrLinkIcon. Have the user choose the new display mode.)

```

(LET ((Card (NC.CardFromWindow Window))
      (Pointer (NC.FetchPointerFromPointerIcon PointerIcon))
      (DestinationCard GlobalLinkFlg OldDisplayMode)
      (SETQ OldDisplayMode (fetch (NCPointer DisplayMode) of Pointer)))
  (if (NC.CheckForNotReadOnly Card Window "Can't change display mode for pointers in ")
    then [COND
          ((WINDOWP Window))
          ((NC.ActiveCardP Card)
           (SETQ Window (NC.FetchWindow Card))

```

(\* Get new display mode from user)

```

      (SETQ DestinationCard (fetch (NCPointer DestinationCard) of Pointer))
      (NC.IfCardPartNotBusy DestinationCard 'FROMLINKS
       (if (NOT NewDisplayMode)
         then (SETQ NewDisplayMode
                (NC.DisplayModeFromStylesheetSelections
                 (STYLESHEET (CREATE.STYLE 'ITEMS [LIST (create MENU
                                                         ITEMS _ '(Yes No Float))
                                                         (create MENU
                                                         ITEMS _ '(Yes No Float))
                                                         (create MENU
                                                         ITEMS _ '(Yes No Float)
                                                         'ITEM.TITLES
                                                         '(Title?% LinkType?% AttachBitmap?)
                                                         'ITEM.TITLE.FONT
                                                         (FONTCOPY MENUFONT 'WEIGHT 'BOLD)
                                                         'SELECTIONS
                                                         (NC.StylesheetSelectionsFromDisplayMode OldDisplayMode)

```

```

'NEED.NOT.FILL.IN NIL 'TITLE 'Display% Mode? 'POSITION
(create POSITION
XCOORD _ LASTMOUSEX
YCOORD _ LASTMOUSEY]
(if (OR (NULL NewDisplayMode)
(EQUAL NewDisplayMode OldDisplayMode))
then
(* * If new display mode is same as old, then bail out.)
NIL
else
(* * Update link in PointerIcon or in global links list)
(replace (NCPointer DisplayMode) of Pointer with NewDisplayMode)
(NC.MarkCardDirty Card) (* This causes tedit to redisplay the screen.)
'CHANGED])

```

**(NC.RelabelPointer**

[LAMBDA (PointerIcon Window NewLinkLabel ForceRedisplayFlg) (\* rht%: "25-Oct-87 19:37" pp pp)

(\* \* Relabel a NoteCard pointer. Ask user for new label.)

```

(LET ((Pointer (NC.FetchLinkFromLinkIcon PointerIcon))
(Card (NC.CardFromWindow Window)
LinkIcon DestinationCard OldLabel)
(if (NC.CheckForNotReadOnly Card Window "Can't change link type of pointers in ")
then [COND
((WINDOWP Window))
((NC.ActiveCardP Card)
(SETQ Window (NC.FetchWindow Card)
(SETQ DestinationCard (fetch (NCPointer DestinationCard) of Pointer))
(NC.IfCardPartNotBusy DestinationCard 'FROMLINKS (COND
((NC.SystemLinkLabelP (fetch (NCPointer Label
)
of Pointer))
(NC.PrintMsg Window T "This is a system
maintained pointer." (CHARACTER
13)
"You cannot change its label."
(CHARACTER 13))
(SPAWN.MOUSE)
(DISSMISS 1500)
(NC.ClearMsg Window T))
((SETQ NewLinkLabel
(OR NewLinkLabel
(NC.AskLinkLabel Window NIL NIL T T)
))

```

(\* \* Put new label in Pointer Icon or global links list)

```

(replace (NCPointer Label) of Pointer
with NewLinkLabel)
(NC.MarkCardDirty Card)
'CHANGED])

```

**(NC.ChangeCardTitleFromPointerIcon**

[LAMBDA (PointerIcon Window)

; Edited 3-Dec-87 19:00 by rht:

(\* \* From a pointer icon sitting in Window, change the title of the pointer's destination card.)

```

(LET (Pointer Card DestinationCard OldTitle NewTitle)
(RESETLST
[RESETSAVE NIL `(MOVEW ,Window , (LOWERLEFT (WINDOWPROP Window 'REGION)
(if (NC.PointerIconImageObjP PointerIcon)
then
(SETQ Pointer (NC.FetchPointerFromPointerIcon PointerIcon))
(SETQ Card (NC.CardFromWindow Window))
(SETQ DestinationCard (fetch (NCPointer DestinationCard) of Pointer))
(if (NC.CheckForNotReadOnly Card Window "Can't change titles for cards in ")
then (if (NC.CrossFileLinkCardP DestinationCard)
then (NC.PrintMsg Window T "Can't change card title from a cross notefile link.")
(DISSMISS 1500)
(NC.ClearMsg Window T)
NIL
else (NC.IfCardPartNotBusy
DestinationCard
'TITLE
(if (SETQ NewTitle (NC.AskUser (CONCAT "Enter new title for card with title '"
(NC.RetrieveTitle DestinationCard)
"' "
(CHARACTER 13) )

```

```
(if (AND (STREQUAL (SETQ OldTitle (NC.RetrieveTitle
DestinationCard))
"Untitled")
(NC.FetchNewCardFlg DestinationCard))
then NIL
else OldTitle)
T Window))
then (NC.AssignTitle DestinationCard NIL NewTitle)
'CHANGED]]))
```

)

(DEFINEQ

**(NC.DeleteBadPointer**

```
[LAMBDA (Pointer ImageObj ImageStream StreamType TextStream Scale)
(* rht%: "24-Oct-87 19:04")
```

(\* \* Deletes a bad pointer when it is discovered at display time.)

```
(LET (Card)
(NC.ReplaceWithDeletedLinkImageObj ImageObj)
(APPLY* (IMAGEOBJPROP ImageObj 'DISPLAYFN)
ImageObj ImageStream StreamType TextStream Scale)
'CHANGED])
```

**(NC.DrawLinkOrPointerIcon**

```
[LAMBDA (ImageStream ImageBox LinkIconString AttachBitmapFlg Card PointerStyleFlg BorderWidthIncrement)
; Edited 10-Jan-89 17:24 by rtk
```

;; Draw the icon representing either a link or a pointer object.

;; pmi 11/3/87: Added ability to print new cross-file link icon, with bitmap attached to right side of link icon.

;; rht 1/21/88: Now takes extra BorderWidthIncrement argument which gets added in to calc of ScaledBorderWidth.

;; pmi 2/3/88: Did minor tweak to part that notices that the icon(s) won't fit in the width defined by NC.LinkIconMaxWidth. Added MultiLineIconFlg, which is set when we have boundary condition of title fitting on one line ONLY if we use the multi-line attached bitmap.

;; pmi 9/28/88: NC.FetchCrossFileLinkIconAttachedBitMap has become obsolete, so now calls NC.FetchLinkIconAttachedBitMap instead.

;; bk 1/5/89: Default Scale to 1

;; bk 1/10/89: Add CachedCardType to call to NC.FetchLinkIconAttachedBitMap for crossfile links.

```
(DECLARE (GLOBALVARS NC.LinkIconMultiLineMode NC.LinkIconMaxWidth NC.LinkIconSpaceInnerX
NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerY NC.LinkIconSpaceOuterY
NC.LinkIconSpaceInterLine NC.LinkIconBorderWidth NC.PointerIconDashingStyle))
(PROG ((Scale (OR (DSPSCALE NIL ImageStream)
1))
(Font (DSPFONT NIL ImageStream))
Dashing FontDescent FontAscent FontHeight Left Bottom Top CrossFileLinkFlg ActualCard Icon SavedIcon
CrossFileLinkIcon LinkIconStrings ApproxBoxWidth ApproxBoxHeight MultiLineIconFlg BoxWidth BoxHeight
ScaledIconHeight ScaledIconWidth XPosition ScaledLinkIconMaxWidth ScaledBorderWidth ScaledSpaceInnerX
ScaledSpaceOuterX ScaledSpaceInnerY ScaledSpaceOuterY ScaledSpaceInterLine HalfScaledSpaceInterLine
BottomOfLine CachedCardType)
```

;; Make temporaries of scaled variables

```
(SETQ ScaledLinkIconMaxWidth (TIMES Scale NC.LinkIconMaxWidth))
(SETQ ScaledBorderWidth (TIMES Scale (PLUS (OR (NUMBERP BorderWidthIncrement)
0)
NC.LinkIconBorderWidth)))
(SETQ ScaledSpaceInnerX (TIMES Scale NC.LinkIconSpaceInnerX))
(SETQ ScaledSpaceOuterX (TIMES Scale NC.LinkIconSpaceOuterX))
(SETQ ScaledSpaceInnerY (TIMES Scale NC.LinkIconSpaceInnerY))
(SETQ ScaledSpaceOuterY (TIMES Scale NC.LinkIconSpaceOuterY))
(SETQ ScaledSpaceInterLine (TIMES Scale NC.LinkIconSpaceInterLine))
(SETQ HalfScaledSpaceInterLine (IQUOTIENT ScaledSpaceInterLine 2))
(SETQ FontDescent (FONTDESCENT Font))
(SETQ FontAscent (FONTASCENT Font))
```

;; FONTHEIGHT doesn't give us the maximum possilbe height of a font, so use (PLUS FontDescent FontAscent) instead.

```
(SETQ FontHeight (PLUS FontDescent FontAscent))
```

;; Indicate Pointer vs Link using dashing style.

```
(if PointerStyleFlg
then (SETQ Dashing NC.PointerIconDashingStyle))
```

;; Set up the icon, if displayed

```
(if AttachBitmapFlg
then ;; Attached icon
;; Check to see if this link points to a card in another notefile.
(if (EQ (NCP.CardType Card)
'CrossFileLink)
then (SETQ CrossFileLinkFlg T)
(SETQ ActualCard (NCP.GetCrossFileLinkDestCard Card NIL T))
(if ActualCard
then (SETQ CachedCardType (NCP.CardType ActualCard))
(NCP.CardProp Card 'CachedCardType CachedCardType))
```

```

        else (SETQ CachedCardType (NCP.CardProp Card 'CachedCardType]
    else (SETQ CrossFileLinkFlg NIL)
        (SETQ ActualCard Card))
;; Use an estimate of the width and height to tell if the box contains more than one line of text
[if NC.LinkIconMultiLineMode
    then ;; Multi-line link icons are enabled
        (SETQ ApproxBoxWidth (PLUS ScaledSpaceOuterX ScaledBorderWidth ScaledSpaceInnerX
            (STRINGWIDTH LinkIconString Font)
            ScaledSpaceInnerX ScaledBorderWidth ScaledSpaceOuterX))
        (SETQ ApproxBoxHeight (PLUS ScaledSpaceOuterY ScaledBorderWidth ScaledSpaceInnerY
            FontHeight ScaledSpaceInnerY ScaledBorderWidth
            ScaledSpaceOuterY))
        ;; Use the box height to determine if Multi-line or Single line. Calculate the correct bitmap.
        (SETQ BoxHeight (fetch (IMAGEBOX YSIZE) of ImageBox))
        (if (GREATERP BoxHeight ApproxBoxHeight)
            then ;; Calculate height for Multi-line icon
                (SETQ MultiLineIconFlg T)
                (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                    HalfScaledSpaceInterLine))
                (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale
                    CachedCardType))
                (SETQ SavedIcon (NC.FetchLinkIconAttachedBitMap ActualCard
                    (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                    ScaledSpaceInnerY ScaledBorderWidth)
                    Scale CachedCardType))
                else ;; Calculate height for Single line icon
                    (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                        ScaledSpaceInnerY ScaledBorderWidth))
                    (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale
                        CachedCardType))
                    (SETQ SavedIcon Icon))
                    (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
                    (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
            else ;; Multi-line link icons are disabled
                (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                    ScaledSpaceInnerY ScaledBorderWidth))
                (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale
                    CachedCardType))
                (SETQ SavedIcon Icon)
                (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
                (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
                (SETQ CrossFileLinkIcon (if CrossFileLinkFlg
                    then (NC.FetchLinkIconAttachedBitMap Card ScaledIconHeight Scale
                        CachedCardType)
                    else NIL))
        else ;; No attached icon
            (SETQ Icon NIL)
            (SETQ SavedIcon Icon)
            (SETQ CrossFileLinkIcon NIL)
            (SETQ ScaledIconHeight 0)
            (SETQ ScaledIconWidth 0))
;; Now determine the text to be printed
(if (AND Icon (GREATERP (PLUS ScaledSpaceInnerX ScaledSpaceInnerX ScaledIconWidth
    (if CrossFileLinkIcon
        then ScaledIconWidth
        else ScaledBorderWidth))
    ScaledLinkIconMaxWidth))
    then ;; If the width of the icon(s)/borders plus the inner x's is greater than the ScaledLinkIconMaxWidth set by the user, set the icons
        ;; and text to NIL
        (SETQ Icon NIL)
        (SETQ SavedIcon Icon)
        (SETQ CrossFileLinkIcon NIL)
        (SETQ ScaledIconHeight 0)
        (SETQ ScaledIconWidth 0)
        (SETQ LinkIconStrings (LIST ""))
    elseif (STREQUAL LinkIconString "")
        then ;; There is no text to print, so set it to the null string
            (SETQ LinkIconStrings (LIST ""))
        else ;; Have the text parsed into separate lines
            (SETQ LinkIconStrings (NC.CreateLinkIconStrings LinkIconString Icon CrossFileLinkIcon ImageStream))
    )
;; Compute all the size values.
(SETQ Bottom (PLUS (DIFFERENCE (DSPYPOSITION NIL ImageStream)
    (fetch (IMAGEBOX YDESC) of ImageBox))
    ScaledSpaceOuterY))

```



```
(SETQ BoxHeight (DIFFERENCE (fetch (IMAGEBOX YSIZE) of ImageBox)
                             (PLUS ScaledSpaceOuterY ScaledSpaceOuterY)))
(SETQ Top (PLUS Bottom BoxHeight))
(SETQ Left (PLUS (DSPXPOSITION NIL ImageStream)
                ScaledSpaceOuterX))
(SETQ BoxWidth (DIFFERENCE (fetch (IMAGEBOX XSIZE) of ImageBox)
                           (PLUS ScaledSpaceOuterX ScaledSpaceOuterX)))
```

;; Display the icon, if possible

```
(if (AND Icon (LEQ (PLUS (if CrossFileLinkIcon
                        then ScaledIconWidth
                        else ScaledBorderWidth)
                    ScaledIconWidth ScaledSpaceInnerX ScaledSpaceInnerX)
                 ScaledLinkIconMaxWidth))
    then ;; Put out the icon bitmap, but only if the width of the icon(s)/borders plus the inner x's does not exceed the LinkIconMaxWidth
        ;; set by the user.
        (BITBLT Icon 0 0 ImageStream Left (DIFFERENCE Top ScaledIconHeight))
        ;; DSPFONT is a kludge to get around bug in PRESS BITBLT which sets the width of a space char to NIL. Bug reported as AR
        ;; #5630.0
        (DSPFONT Font ImageStream)
        (if (ZEROP (NCHARS LinkIconString))
            then ;; Display the cross-file link icon, if necessary
                (if CrossFileLinkIcon
                    then (BITBLT CrossFileLinkIcon 0 0 ImageStream (DIFFERENCE (PLUS Left BoxWidth)
                                         ScaledIconWidth)
                                (DIFFERENCE Top ScaledIconHeight)))
                    ;; Quit because just a typeicon
                    (RETURN))
                else ;; There isn't enough space for the icon, so set it to NIL
                    (SETQ Icon NIL)
                    (SETQ ScaledIconWidth 0))
        ;; Enter the appropriate text.
```

```
(SETQ XPosition (PLUS Left (if Icon
                            then ScaledIconWidth
                            else ScaledBorderWidth)
                    ScaledSpaceInnerX))
(DSPXPOSITION XPosition ImageStream)
(if LinkIconStrings
    then (if (OR (GREATERP (LENGTH LinkIconStrings)
                          1)
                MultiLineIconFlg)
            then (SETQ BottomOfLine (DIFFERENCE Top (PLUS ScaledBorderWidth ScaledSpaceInnerY
                                                         FontAscent)))
                (SETQ XPosition (PLUS Left ScaledSpaceInnerX ScaledBorderWidth))
                (for String in LinkIconStrings do (DSPYPOSITION BottomOfLine ImageStream)
                (PRIN1 String ImageStream)
                (SETQ BottomOfLine (DIFFERENCE BottomOfLine
                                                (PLUS FontDescent
                                                  ScaledSpaceInterLine
                                                  FontAscent)))
                (DSPXPOSITION XPosition ImageStream))
            ;; Draw the box for multiple lines of text.
            (NC.DrawInnerBox Left Bottom BoxWidth BoxHeight ScaledBorderWidth NIL ImageStream
                             Dashing Icon CrossFileLinkIcon ScaledIconWidth ScaledIconHeight)
            else (DSPYPOSITION (PLUS Bottom ScaledBorderWidth ScaledSpaceInnerY FontDescent)
                               ImageStream)
                 (PRIN1 (CAR LinkIconStrings)
                        ImageStream)
            ;; Draw the box for a single line of text.
            (NC.DrawInnerBox (PLUS Left ScaledIconWidth)
                             Bottom
                             (DIFFERENCE BoxWidth (if CrossFileLinkIcon
                                                         then (PLUS ScaledIconWidth ScaledIconWidth)
                                                         else ScaledIconWidth))
                             BoxHeight ScaledBorderWidth NIL ImageStream Dashing Icon CrossFileLinkIcon)))
```

;; Display the cross-file link icon, if necessary

```
(if CrossFileLinkIcon
    then (BITBLT CrossFileLinkIcon 0 0 ImageStream (DIFFERENCE (PLUS Left BoxWidth)
                                                                ScaledIconWidth)
          (DIFFERENCE Top ScaledIconHeight))
```

**(NC.ComputeLinkOrPointerImageBox**

```
[LAMBDA (ImageStream LinkIconString AttachBitmapFlg Card RightMargin BorderWidthIncrement CurrentX)
    ; Edited 6-Jan-89 14:17 by rtk
```

;; Compute the size of the image box needed to print a link or pointer icon.

;; pmi 11/3/87: Added ability to print new cross-file link icon, with bitmap attached to right side of link icon.

```

;; rht 1/21/88: Now takes extra BorderWidthIncrement argument which gets added in to calc of ScaledBorderWidth.
;; pmi 2/3/88: Did minor tweak to various things. Added MultiLineIconFlg, which is set when we have boundary condition of title fitting on one line
;; ONLY if we use the multi-line attached bitmap.
;; pmi 9/28/88: NC.FetchCrossFileLinkIconAttachedBitMap has become obsolete, so now calls NC.FetchLinkIconAttachedBitMap instead.
;; BK 1/6/89 Add Default Scale of 1
;; Sye Jan/18/89 make sure YDESC is non-zero only when RightMargin is none-nil and the window ImageStream associated with has a
;; TEXTSTREAM
(DECLARE (GLOBALVARS NC.LinkIconMultiLineMode NC.LinkIconMaxWidth NC.LinkIconBorderWidth
          NC.LinkIconSpaceInnerX NC.LinkIconSpaceOuterX NC.LinkIconSpaceInnerY NC.LinkIconSpaceOuterY
          NC.LinkIconSpaceInterLine))
(LET ((Scale (OR (DSPSCALE NIL ImageStream)
                1))
      (BoxWidth 0)
      (Font (DSPFONT NIL ImageStream))
      CrossFileLinkFlg ActualCard Icon SavedIcon CrossFileLinkIcon FontHeight ApproxBoxWidth
      MultiLineIconFlg LinkIconStrings ScaledIconHeight ScaledIconWidth NumberOfLines XSize YSize
      ScaledLinkIconMaxWidth ScaledBorderWidth ScaledSpaceInnerX ScaledSpaceOuterX ScaledSpaceInnerY
      ScaledSpaceOuterY ScaledSpaceInterLine HalfScaledSpaceInterLine TotalEdgeSpaceY)
      ;; FONTHEIGHT doesn't give us the maximum possible height of a font, so use (PLUS FontDescent FontAscent) instead.
      (SETQ FontHeight (PLUS (FONTDESCENT Font)
                            (FONTASCENT Font)))
      ;; Make temporaries of scaled vars.
      (SETQ ScaledLinkIconMaxWidth (TIMES Scale NC.LinkIconMaxWidth))
      (SETQ ScaledBorderWidth (TIMES Scale (PLUS (OR (NUMBERP BorderWidthIncrement)
                                                    0)
                                                NC.LinkIconBorderWidth)))
      (SETQ ScaledSpaceInnerX (TIMES Scale NC.LinkIconSpaceInnerX))
      (SETQ ScaledSpaceOuterX (TIMES Scale NC.LinkIconSpaceOuterX))
      (SETQ ScaledSpaceInnerY (TIMES Scale NC.LinkIconSpaceInnerY))
      (SETQ ScaledSpaceOuterY (TIMES Scale NC.LinkIconSpaceOuterY))
      (SETQ ScaledSpaceInterLine (TIMES Scale NC.LinkIconSpaceInterLine))
      (SETQ HalfScaledSpaceInterLine (IQUOTIENT ScaledSpaceInterLine 2))

      (**)
      (SETQ TotalEdgeSpaceY (PLUS ScaledBorderWidth ScaledSpaceOuterY ScaledSpaceInnerY))
      (SETQ YSize (PLUS TotalEdgeSpaceY TotalEdgeSpaceY FontHeight))
      ;; Set up the icon, if displayed
      (if AttachBitmapFlg
          then ;; Attached icon
              ;; Check to see if this link points to a card in another notefile.
              (if (EQ (NCP.CardType Card)
                    'CrossFileLink)
                  then (SETQ CrossFileLinkFlg T)
                      (SETQ ActualCard (NCP.GetCrossFileLinkDestCard Card NIL T))
                  else (SETQ CrossFileLinkFlg NIL)
                      (SETQ ActualCard Card))
              ;; Use an estimate of the width to tell if the box contains more than one line of text
              [if NC.LinkIconMultiLineMode
                then ;; Multi-line link icons are enabled
                    (SETQ ApproxBoxWidth (PLUS ScaledSpaceInnerX (STRINGWIDTH LinkIconString Font)
                                                ScaledSpaceInnerX))
                    ;; The image box must be calculated
                    (if (GREATERP ApproxBoxWidth ScaledLinkIconMaxWidth)
                        then ;; Calculate Multi-line icon
                            (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                                                         HalfScaledSpaceInterLine))
                            (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale)
                                      )
                            (SETQ SavedIcon Icon)
                            (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
                            (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
                        else ;; Calculate Single line icon
                            (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                                                         ScaledSpaceInnerX))
                            (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale))
                            (SETQ SavedIcon Icon)
                            (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
                            (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
                            ;; Now see if total width, including the icon, will still fit if this is a Single line
                            (if (AND (NOT (STREQUAL LinkIconString ""))
                                     (GREATERP (PLUS ApproxBoxWidth ScaledIconWidth
                                                       (if CrossFileLinkFlg
                                                           then ScaledIconWidth
                                                           else ScaledBorderWidth))
                                     YSize))
                                then
                                    (SETQ ScaledIconWidth (PLUS ScaledIconWidth ScaledBorderWidth))
                                else
                                    (SETQ ScaledIconWidth (PLUS ScaledIconWidth ScaledBorderWidth))
                                )
                            )
                    )
                )
            )
    )

```

```

        ScaledLinkIconMaxWidth))
    then ;; Now it doesn't fit. Calculate Multi-line icon
        (SETQ MultiLineIconFlg T)
        (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY
                                     FontHeight HalfScaledSpaceInterLine))
        (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight
                                                    Scale))
        (SETQ SavedIcon (NC.FetchLinkIconAttachedBitMap ActualCard
                                                         (PLUS ScaledBorderWidth ScaledSpaceInnerY
                                                             FontHeight ScaledSpaceInnerY
                                                             ScaledBorderWidth)
                                                         Scale))
        (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
        (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
    else ;; Multi-line link icons are disabled
        (SETQ ScaledIconHeight (PLUS ScaledBorderWidth ScaledSpaceInnerY FontHeight
                                     ScaledSpaceInnerY ScaledBorderWidth))
        (SETQ Icon (NC.FetchLinkIconAttachedBitMap ActualCard ScaledIconHeight Scale))
        (SETQ SavedIcon Icon)
        (SETQ ScaledIconHeight (TIMES Scale (BITMAPHEIGHT Icon)))
        (SETQ ScaledIconWidth (TIMES Scale (BITMAPWIDTH Icon)))
        (SETQ CrossFileLinkIcon (if CrossFileLinkFlg
                                   then (NC.FetchLinkIconAttachedBitMap Card ScaledIconHeight Scale)
                                   else NIL))
    else ;; No attached icon
        (SETQ Icon NIL)
        (SETQ SavedIcon Icon)
        (SETQ CrossFileLinkIcon NIL)
        (SETQ ScaledIconHeight 0)
        (SETQ ScaledIconWidth 0)
;; Now determine the actual size of the image box
[if (AND Icon (GREATERP (PLUS ScaledSpaceInnerX ScaledSpaceInnerX ScaledIconWidth
                            (if CrossFileLinkIcon
                                then ScaledIconWidth
                                else ScaledBorderWidth))
                        ScaledLinkIconMaxWidth))
    then ;; If the width of the icon plus the width of the cross-file link icon (if any) plus the inner x's is greater than the
        ;; ScaledLinkIconMaxWidth set by the user, set the box width to ScaledLinkIconMaxWidth plus the width of the outer x's.
        (SETQ XSize (PLUS ScaledSpaceOuterX ScaledSpaceOuterX ScaledLinkIconMaxWidth))
    elseif (ZEROP (NCHARS LinkIconString))
    then (if Icon
          then ;; There is no text to print, so the box width is just the icon width plus the width of the cross-file link icon (if any) plus
              ;; the outer x on each side
              (SETQ XSize (PLUS ScaledSpaceOuterX ScaledSpaceOuterX ScaledIconWidth
                              (if CrossFileLinkIcon
                                  then ScaledIconWidth
                                  else ScaledBorderWidth)))
          else ;; There is no text or icon to print, so the box width is just the inner x, border width, and outer x on each side
              (SETQ XSize (PLUS ScaledSpaceInnerX ScaledSpaceInnerX ScaledBorderWidth
                              ScaledBorderWidth ScaledSpaceOuterX ScaledSpaceOuterX)))
    else ;; Have the text parsed into separate lines
        (SETQ LinkIconStrings (NC.CreateLinkIconStrings LinkIconString Icon CrossFileLinkIcon ImageStream))
        (if LinkIconStrings
            then (SETQ NumberOfLines (LENGTH LinkIconStrings))
                [if (GREATERP NumberOfLines 1)
                    then (SETQ BoxWidth (PLUS (STRINGWIDTH (CAR LinkIconStrings)
                                                    Font)
                                             (if Icon
                                                 then ScaledIconWidth
                                                 else ScaledBorderWidth)
                                             (if CrossFileLinkIcon
                                                 then ScaledIconWidth
                                                 else ScaledBorderWidth)))
                    ;; Find the longest string for the width of the box
                    (for String in (CDR LinkIconStrings) bind PartStringWidth
                        when (GREATERP (SETQ PartStringWidth (PLUS (STRINGWIDTH String Font)
                                                                    ScaledBorderWidth
                                                                    ScaledBorderWidth))
                                     BoxWidth)
                            do (SETQ BoxWidth PartStringWidth))
                    (SETQ YSize (PLUS TotalEdgeSpaceY TotalEdgeSpaceY (TIMES NumberOfLines
                                                                                FontHeight)
                                     (TIMES (SUB1 NumberOfLines)
                                             ScaledSpaceInterLine)))
                else ;; It's possible to have a string that won't fit on one line with a single line icon, but will fit if we use the multi-line
                    ;; icon. In this case, we have to fool it into thinking it has another line.
                    [if (AND MultiLineIconFlg (EQ NumberOfLines 1))

```

```

    then (SETQ NumberOfLines (ADD1 NumberOfLines))
          (SETQ YSize (PLUS TotalEdgeSpaceY TotalEdgeSpaceY (TIMES NumberOfLines
                                                                    FontHeight)
                                                                    (TIMES (SUB1 NumberOfLines)
                                                                    ScaledSpaceInterLine]
          (SETQ BoxWidth (PLUS (STRINGWIDTH (CAR LinkIconStrings)
                                          Font)
                              (if Icon
                                  then ScaledIconWidth
                                  else ScaledBorderWidth)
                              (if CrossFileLinkIcon
                                  then ScaledIconWidth
                                  else ScaledBorderWidth]
          (SETQ XSize (MIN (PLUS BoxWidth ScaledSpaceOuterX ScaledSpaceInnerX
                              ScaledSpaceInnerX)
                          (PLUS ScaledLinkIconMaxWidth ScaledSpaceOuterX ScaledSpaceOuterX]
    (create IMAGEBOX
            XSIZE _ XSize
            YSIZE _ YSize
            YDESC _ (COND
                    [RightMargin (COND
                                (= Scale 1)
                                ;; this is a display stream
                                (if (ZEROP CurrentX)
                                    then 0
                                    else (PLUS (FONTDESCENT Font)
                                                TotalEdgeSpaceY)))
                                (T ;; this is a print stream
                                 (if (AND (ZEROP (DSPLEFTMARGIN NIL ImageStream))
                                         (ZEROP (DSPBOTTOMMARGIN NIL ImageStream)))
                                     then 0
                                     else (PLUS (FONTDESCENT Font)
                                                TotalEdgeSpaceY]
                    (T 0))
            XKERN _ 0))
    )

```

(DECLARE%: DONTEVAL@LOAD

(NC.MakePointerIcon)

;;; New stuff for allowing put to work from TEdit.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.ExternalPutLinkIconImageFns)

```

(RPAQ? NC.ExternalPutLinkIconImageFns (IMAGEFNSCREATE (FUNCTION NC.ExternalPutLinkIconDisplayFn)
                                                       (FUNCTION NC.ExternalPutLinkIconImageBoxFn)
                                                       (FUNCTION NC.ExternalPutLinkIconPutFn)
                                                       (FUNCTION NC.ExternalPutLinkIconGetFn)
                                                       NIL
                                                       (FUNCTION NIL)))

```

(DEFINEQ

```

(NC.ExternalPutLinkIconDisplayFn
 [LAMBDA (ImageObj ImageStream STREAMTYPE TEXTSTREAM SCALE) ; Edited 5-Jan-89 16:33 by rtk

```

;;; bk 1/5/89 Check for a LINKBITMAP

(\* \* Display a link icon)

```

(PROG ((Font (FONTCREATE 'HELVETICA 10 'BOLD))
      (DEST-BITMAP (IMAGEOBJPROP ImageObj 'IMAGEBITMAP))
      LINK-BOX)
      (if DEST-BITMAP
          then (SETQ LINK-BOX (IMAGEOBJPROP ImageObj 'IMAGEBOX))
               (BITBLT DEST-BITMAP 0 0 ImageStream (DSPXPOSITION NIL ImageStream)
                       (DSPYPOSITION NIL ImageStream)
                       (fetch (IMAGEBOX XSIZE) of LINK-BOX)
                       (fetch (IMAGEBOX YSIZE) of LINK-BOX)
                       'INPUT
                       'REPLACE)
          else (DSPFONT (PROG1 (DSPFONT Font ImageStream)
                              (PRIN1 (IMAGEOBJPROP ImageObj 'OBJECTDATUM)
                                       ImageStream))
                    ImageStream])

```

(NC.ExternalPutLinkIconImageBoxFn

[LAMBDA (ImageObj ImageStream CurrentX RightMargin) ; Edited 12-Jan-89 17:08 by rtk

;;; bk 1/5/89 look for IMAGEBOX property

;;; bk 1/11/89 Check for a valid link obj before making a bitmap.

```
(PROG ((FONT (FONTCREATE 'HELVETICA 10 'BOLD))
      (Scale (DSPSCALE NIL ImageStream)))
  (RETURN (OR (IMAGEOBJPROP ImageObj 'IMAGEBOX)
             (AND (NC.ValidLinkP (FETCH (IMAGEOBJ OBJECTDATUM) OF ImageObj))
                  (CL:MULTIPLE-VALUE-BIND (DEST-BITMAP DEST-IMAGEBOX)
                                           (NC.SetBitMap&Box ImageObj)
                                           DEST-IMAGEBOX))
             (create IMAGEBOX
                     XSIZE _ (TIMES Scale (STRINGWIDTH (IMAGEOBJPROP ImageObj 'OBJECTDATUM)
                                                         FONT))
                     YSIZE _ (TIMES Scale (FONTPROP FONT 'HEIGHT))
                     YDESC _ (COND
                              (RightMargin ; This is in a TEdittextstream
                               (FONTPROP FONT 'DESCENT))
                              (T 0))
                     XKERN _ 0))
```

**(NC.ExternalPutLinkIconPutFn**

[LAMBDA (ImageObject Stream) ; Edited 24-Jan-89 15:54 by rtk

;;; bk 1/5/89 Write the BitMap Object if one exists (assumes the GETFN was changed to BMOBJ.GETFN3.

;;; bk 1/11/89 Check for a valid link obj before making a bitmap.

;;; bk 1/24/89 Adempt to get right YDESC for textstreams by calling NC.LinkIconImageBoxFn again.

;;; Writes the Link information in the link icon specified by ImageObject onto the file specified by Stream.

```
(LET (String Link DEST-BITMAP LINK-BOX)
  (COND
   ((NC.ValidLinkP (FETCH (IMAGEOBJ OBJECTDATUM) OF ImageObject))
    (LET* ([DEST-BITMAP (NC.SetBitMap&Box ImageObject (IMAGEOBJPROP ImageObject 'IMAGEBITMAP)
                                         (TEXT-IMAGEBOX (NC.LinkIconImageBoxFn ImageObject Stream 1 0))
                                         (TEXT-DESC (AND TEXT-IMAGEBOX (fetch (IMAGEBOX YDESC) of TEXT-IMAGEBOX)))
                                         [BOUNDBOX-DESC (AND (IMAGEOBJPROP ImageObject 'BOUNDBOX)
                                                             (fetch (IMAGEBOX YDESC) of (IMAGEOBJPROP ImageObject 'BOUNDBOX))
                                         [IMAGEBOX-DESC (AND (IMAGEOBJPROP ImageObject 'IMAGEBOX)
                                                             (fetch (IMAGEBOX YDESC) of (IMAGEOBJPROP ImageObject 'IMAGEBOX))
                                         (BITMAP-OBJECT (BITMAPEDITOBJ DEST-BITMAP NIL NIL (OR TEXT-DESC BOUNDBOX-DESC IMAGEBOX-DESC
                                                                                                     ]
                                         (APPLY* (IMAGEOBJPROP (BITMAPEDITOBJ)
                                                             'PUTFN)
                                         BITMAP-OBJECT Stream)))
    ([SETQ String (STRINGP (IMAGEOBJPROP ImageObject 'OBJECTDATUM)
                          (PRIN2 String Stream))
      (T (SETQ Link (NC.FetchLinkFromLinkIcon ImageObject))
         (PRIN2 (CONCAT "****[" (fetch (Link Label) of Link)
                        " link to "
                        (NC.RetrieveTitle (fetch (Link DestinationCard) of Link)
                        "]]****")
                  Stream))
```

**(NC.ExternalPutLinkIconGetFn**

[LAMBDA (Stream) (\* fgh%: "30-May-85 16:43")

(\* \* Reads the link information from Stream and returns a link icon image object for this link info.)

```
(IMAGEOBJCREATE (READ Stream)
  NC.ExternalPutLinkIconImageFns])
```

**(NC.ExternalGetLinkIconFn**

[LAMBDA (STREAM) ; Edited 28-Feb-89 11:32 by krivacic

;;; Call The Latest BITMAP GETFN to read the image object back

```
(APPLY* (IMAGEOBJPROP (BITMAPEDITOBJ)
  'GETFN)
  STREAM])
```

**(NC.CoerceToExternalPutLinkIcon**

[LAMBDA (ImageObject SourceCard DestinationCard LinkLabel) ; Edited 5-Jan-89 18:36 by rtk

;;; Make image object into an external put link icon with the given data

;;; kirk 15Nov85: deleted use of LinkID

;;; bk 1/5/89 Make the ObjectDatum be a bitmap so it can be displayed

```
(LET* [(ORIGINAL-IMAGE-OBJ (IMAGEOBJPROP ImageObject 'OriginalImageObject))
      (DEST-BITMAP (AND ORIGINAL-IMAGE-OBJ (NC.LinkToBitMap ORIGINAL-IMAGE-OBJ)
      (if DEST-BITMAP
        then (NC.SetBitMap&Box ImageObject DEST-BITMAP)
              (RPLACA (IMAGEOBJPROP ImageObject 'OBJECTDATUM)
                      DEST-BITMAP)
        else (IMAGEOBJPROP ImageObject 'OBJECTDATUM (CONCAT "[" LinkLabel " link to " (NC.RetrieveTitle
                                                                                               DestinationCard)
                                                                                               "]""))
      (replace (IMAGEOBJ IMAGEOBJFNS) of ImageObject with NC.ExternalPutLinkIconImageFns])
)
```

;;; Stuff to register our image object for intermezzo.

(DEFINEQ

**(NC.RegisterPlaceMarkerImageObj**

[LAMBDA NIL

(\* rht%: "31-May-85 11:40")

(\* \* Cause the PlaceMarker Image Obj to be registered by creating the make place marker function once)

(NC.MakePlaceMarker])

)

(DECLARE%: DONTEVAL@LOAD

**(NC.RegisterPlaceMarkerImageObj)**

)

;;; Stuff for Having BitMap external Link Images

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.ExternalBitMapPutLinkIconImageFns NC.LinkIconBitMapStream)

)

```
(RPAQ? NC.ExternalBitMapPutLinkIconImageFns (IMAGEFNSCREATE (FUNCTION NC.ExternalPutLinkIconDisplayFn)
                                                             (FUNCTION NC.ExternalPutLinkIconImageBoxFn)
                                                             (FUNCTION NC.ExternalPutLinkIconPutFn)
                                                             (FUNCTION NC.ExternalGetLinkIconFn)
                                                             NIL
                                                             (FUNCTION NIL)))
```

(SETQ NC.LinkIconBitMapStream (DSPCREATE (BITMAPCREATE 512 128)))

;;; Stuff for handling link icons in sketches.

(DEFINEQ

**(NC.LinkIconSketchElementP**

[LAMBDA (SketchElement)

(\* rht%: "20-Aug-85 16:23")

(\* \* Return non-nil if SketchElement is a link icon imageobj.)

(NC.LinkIconImageObjP (SKETCH.IMAGEOBJ.OF.ELEMENT SketchElement]))

**(NC.DeleteLinkIconSketchElement**

[LAMBDA (SketchElement CardID Don'tCreateDeletedImageObjFlg) (\* rht%: " 4-Nov-86 21:01")

(\* \* This replaces the link icon for SketchElement from SketchWin with deleted link icon.)

(\* rht 4/30/86%: Changed to simply delete the link icon if NC.UseDeletedLinkIconIndicators is NIL)

(\* rht 11/4/86%: Now takes Don'tCreateDeletedImageObjFlg arg.)

```
(DECLARE (GLOBALVARS NC.DeletedLinkImageFns NC.UseDeletedLinkIconIndicatorsFlg NC.DeletedLinkImageObject))
(LET ((Position (SKETCH.POSITION.OF.ELEMENT SketchElement))
      (SketchSubstance (NC.FetchSubstance CardID))
      (LinkIcon (SKETCH.IMAGEOBJ.OF.ELEMENT SketchElement))
      (Window (NC.FetchWindow CardID))
      Scale)
  (if (AND NC.UseDeletedLinkIconIndicatorsFlg (NOT Don'tCreateDeletedImageObjFlg))
    then (SETQ Scale (if Window
                        then (VIEWER.SCALE Window)
                        else (NC.FetchScale CardID)))
      (create IMAGEOBJ smashing LinkIcon OBJECTDATUM _ (IMAGEOBJPROP NC.DeletedLinkImageObject
                                                                    'OBJECTDATUM)
              IMAGEOBJFNS _ NC.DeletedLinkImageFns)
      (AND Window (SKETCH.ELEMENT.CHANGED SketchSubstance SketchElement Window))
    else (SKETCH.DELETE.ELEMENT SketchElement SketchSubstance T]))
```

)

;;; Icons representing notecards types.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.FileBoxIcon NC.GraphCardIcon NC.SketchCardIcon NC.TextCardIcon NC.TypelessIcon)  
)

;;; DELETEME imageobjs

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.DELETEMEImageObjDatum NC.RemoveDELETEMEImageObjsFromCardFlg)  
)

(RPAQ? **NC.DELETEMEImageObjDatum** "[[DELETE ME]]")

(RPAQ? **NC.RemoveDELETEMEImageObjsFromCardFlg** NIL)

(DEFINEQ

**(NC.DELETEMEImageObjP**

[LAMBDA (ImageObj)

(\* rht%: " 7-Oct-86 16:26")

(\* \* Returns non-nil if ImageObj is a DELETEME imageobj.)

(**DECLARE** (GLOBALVARS NC.DELETEMEImageObjDatum))

(AND (IMAGEOBJP ImageObj)

(EQUAL (IMAGEOBJPROP ImageObj 'OBJECTDATUM)  
NC.DELETEMEImageObjDatum])

)

(PUTPROPS **NCLINKS FILETYPE** :FAKE-COMPILE-FILE)

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

FUNCTION INDEX

NC.AddFromLink	4	NC.InsertPointerInText	40
NC.AddGlobalLinksToCard	20	NC.InsureLinkDisplayMode	26
NC.AddGlobalLinkToCard	20	NC.InvisibleLinkGetFn	40
NC.AddLinksToCard	24	NC.InvisibleLinkImageBoxFn	40
NC.AddLinkToCard	24	NC.InvisibleLinkPutFn	40
NC.AddLinkToGraphCard	4	NC.LinkAtCharPos	39
NC.AddLinkToSketchCard	4	NC.LinkIconButtonEventInFn	27
NC.AddToLink	5	NC.LinkIconCopyFn	38
NC.CachedAddToLink	6	NC.LinkIconDisplayFn	29
NC.CachedDelFromLink	7	NC.LinkIconGetFn	31
NC.CachedMakeFilingLink	15	NC.LinkIconImageBoxFn	30
NC.ChangeCardTitleFromLinkIcon	20	NC.LinkIconImageObjP	31
NC.ChangeCardTitleFromPointerIcon	46	NC.LinkIconLeftButtonFn	39
NC.ChangeLinkDisplayMode	37	NC.LinkIconMiddleButtonFn	31
NC.ChangeLinkLabelInShowLinksWin	25	NC.LinkIconPutFn	31
NC.ChangePointerDisplayMode	45	NC.LinkIconSketchElementP	54
NC.CheckDisplayModeFormat	26	NC.LinkIconWhenCopiedFn	32
NC.CheckForOrphanDelete	11	NC.LinkIconWhenDeletedFn	33
NC.ChildLinkP	3	NC.LinkIconWhenInsertedFn	34
NC.CoerceToExternalPutLinkIcon	53	NC.LinkIconWhenMovedFn	35
NC.ComputeLinkOrPointerImageBox	49	NC.LinkLabelP	14
NC.ContentsLinkP	3	NC.LinksSupportedP	27
NC.ConvertPointerIconToLinkIcon	41	NC.LinkToBitMap	39
NC.CreateLinkIconStrings	21	NC.MakeChildLink	19
NC.DeleteBadLink	11	NC.MakeFilingLink	16
NC.DeleteBadPointer	47	NC.MakeFilingLinks	16
NC.DeletedLinkDisplayFn	10	NC.MakeGlobalLink	4
NC.DeletedLinkGetFn	10	NC.MakeLink	14
NC.DeletedLinkImageBoxFn	10	NC.MakeLinkIcon	27
NC.DeletedLinkImageObjP	11	NC.MakeNewDisplayMode	26
NC.DeletedLinkPutFn	11	NC.MakePointer	41
NC.DeleteFromLink	9	NC.MakePointerIcon	41
NC.DeleteLink	8	NC.PointerIconCopyFn	43
NC.DeleteLinkIconSketchElement	54	NC.PointerIconDisplayFn	42
NC.DELETEMEImageObjP	55	NC.PointerIconGetFn	43
NC.DeleteToLink	9	NC.PointerIconImageBoxFn	42
NC.DelFromLink	6	NC.PointerIconImageObjP	41
NC.DelReferencesToCard	7	NC.PointerIconLeftButtonFn	44
NC.DelReferencesToCardFromGlobalList	8	NC.PointerIconMiddleButtonFn	44
NC.DelReferencesToCardFromShowLinks	25	NC.PointerIconPutFn	43
NC.DelToLink	7	NC.ReadPointer	44
NC.DisplayModeFromStylesheetSelections	26	NC.RegisterPlaceMarkerImageObj	54
NC.DrawLinkOrPointerIcon	47	NC.RelabelLink	17
NC.EnsureNoCycles	19	NC.RelabelPointer	46
NC.ExternalGetLinkIconFn	53	NC.ReplaceWithDeletedLinkImageObj	11
NC.ExternalPutLinkIconDisplayFn	52	NC.ReverseLinkLabelP	3
NC.ExternalPutLinkIconGetFn	53	NC.SameLinkP	19
NC.ExternalPutLinkIconImageBoxFn	52	NC.SameLinksP	19
NC.ExternalPutLinkIconPutFn	53	NC.SetBitMap&Box	39
NC.FetchLinkFromLinkIcon	36	NC.ShrinkIcon	23
NC.FetchLinkIconForLink	27	NC.StylesheetSelectionsFromDisplayMode	27
NC.FetchPointerFromPointerIcon	42	NC.SubContentsLinkP	3
NC.FetchShowLinksWindow	25	NC.SystemLinkLabelP	19
NC.FileInOrphanBox	12	NC.TraverseLink	21
NC.FillInLinkIcon	40	NC.TraversePointer	45
NC.GlobalLinkP	3	NC.UncacheLinks	21
NC.HookToOrphanCard	12	NC.UpdateLinkImages	38
NC.IDAlphaOrder	14	NC.UpdateLinkImagesInShowLinks	25
NC.InsertLinkBeforeMarker	12	NC.ValidLinkP	19
NC.InsertLinkInSketch	5	NC.WritePointer	44
NC.InsertLinkInText	5		

VARIABLE INDEX

NC.BrowserContentsLinkLabel	3	NC.LinkIconSpaceOuterX	3
NC.DELETEMEImageObjDatum	55	NC.LinkIconSpaceOuterY	3
NC.DocBackPtrLinkLabel	3	NC.LinkIndexBackPtrLinkLabel	3
NC.ExternalBitMapPutLinkIconImageFns	54	NC.ListContentsLinkLabel	3
NC.ExternalPutLinkIconImageFns	52	NC.MenuItems/DisplayModeValues	27
NC.FileBoxLinkSpacer	3	NC.PointerIconDashingStyle	40
NC.FiledCardLinkLabel	3	NC.PointerIconImageObjFns	40
NC.InitialLinkLabels	3	NC.RemoveDELETEMEImageObjsFromCardFlg	55
NC.LinkIconBorderWidth	3	NC.SubBoxLinkLabel	3
NC.LinkIconSpaceInnerX	3	NC.SystemLinkLabels	3
NC.LinkIconSpaceInnerY	3	NC.UCASESystemLinkLabels	3
NC.LinkIconSpaceInterLine	3	NC.UnspecifiedLinkLabel	3



{MEDLEY}<notecards>system>NCLINKS.;1

---

**PROPERTY INDEX**

NCLINKS .....55

---