```
(RPAQQ NCSKETCHCARDCOMS
       (
```

;;; Definition of the Sketch card type.

```
       [DECLARE%: COPY FIRST (P (NC.LoadFileFromDirectories 'SKETCH 'LISPUSERSDIRECTORIES]
       (FNS NCAddStub.SketchCard)
       (GLOBALVARS NC.SketchCardIcon NC.SketchTitleBarMenu NC.DeletedLinkImageObject
              NC.UseDeletedLinkIconIndicatorsFlg)
       (BITMAPS NC.SketchCardIcon)
```

;;; Basic functions

```
       (FNS NC.MakeSketchCard NC.BringUpSketchCard NC.SketchDirtyP NC.SketchTitleBarButtonEventFn
            NC.SketchCardCloseFn NC.SketchCardShrinkFn NC.SketchCopySubstance NC.MarkSketchDirty
            NC.CollectReferencesInSketch NC.GetSketchSubstance NC.PutSketchSubstance NC.PutCachedMap
            NC.GetCachedMap NC.UpdateLinkImagesInSketch NC.DelReferencesToCardFromSketch
            NC.ExternalizeLinkIconsInSketch NC.MakeExternalSketchCopy NC.SK.COPY.BUTTONEVENTFN)
       [INITADVISE (CREATE.SKETCHW.COMMANDMENU (NIL (AFTER NIL (NC.RemoveSketchMenuItems !VALUE]
```

;;; This stuff supports the push-copy method of copying links in NC

```
       (FNS NC.TranslateWindowPositionToSketchPosition)
```

;;; Special stuff for autoloading maps

```
       (FNS NC.FakeMapInputFn NC.FakeMapInsideFn)
       [DECLARE%: DONTEVAL@LOAD (P (IF NIL THEN

                                   ;; Don't load MAP stuff yet

                                   (NC.StoreAutoloadFnFile (FUNCTION SK.MAP.INPUTFN)
                                          'NCMAPS
                                          'NOTECARDSDIRECTORIES)
                                   (NC.StoreAutoloadFnFile (FUNCTION SK.MAP.INSIDEFN)
                                          'NCMAPS
                                          'NOTECARDSDIRECTORIES)
                                   (NC.StoreAutoloadFnFile (FUNCTION SetCachedBitMap)
                                          'NCMAPS
                                          'NOTECARDSDIRECTORIES)
                                   (OR (FMEMB 'MAP SKETCH.ELEMENT.TYPE.NAMES)
                                       (CREATE.SKETCH.ELEMENT.TYPE 'MAP "Map" "Forces autoload of MAP
                                              package." NIL NIL NIL NIL (FUNCTION NC.FakeMapInputFn)
                                              (FUNCTION NC.FakeMapInsideFn]
```

;;; Add sketch card type to CardType list

```
       (FNS NC.AddSketchCard)
       (DECLARE%: DONTEVAL@LOAD (P (NC.AddSketchCard)))
       (PROP (FILETYPE MAKEFILE-ENVIRONMENT)
             NCSKETCHCARD)))
```

;;; Definition of the Sketch card type.

```
(DECLARE%: COPY FIRST

(NC.LoadFileFromDirectories 'SKETCH 'LISPUSERSDIRECTORIES)
)

(DEFINEQ

(NCAddStub.SketchCard
  [LAMBDA NIL                                                      ; Edited  3-Dec-87 18:58 by rht:

          (* * kirk 18Jun86 Add the Sketch card stub)

          (* * rht 11/7/86%: Fixed typo, changing a QUOTE to BQUOTE.)
```

```
      (DECLARE (GLOBALVARS NC.SketchCardIcon))
      (NC.AddCardTypeStub 'Sketch 'NoteCard 'NCSKETCHCARD NIL '((DisplayedInMenuFlg T)
                                                          (LinkIconAttachedBitMap ,NC.SketchCardIcon])
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.SketchCardIcon NC.SketchTitleBarMenu NC.DeletedLinkImageObject NC.UseDeletedLinkIconIndicatorsFlg
       )
)

(RPAQQ NC.SketchCardIcon
```
(RPAQQ **NC.SketchCardIcon**)

```
;;; Basic functions

(DEFINEQ
```

(**NC.MakeSketchCard**
```
  [LAMBDA (Card Title NoDisplayFlg ParamList InterestedWindow RegionOrPosition)
                                                                         ; Edited 14-Dec-88 13:39 by krivacic
```
   ;; rht 2/1/85: Added windowprop to prevent sketch asking about saving changes.

   ;; fgh 11/14/85 Updated to handle Card object.

   ;; rht 5/6/86 Replaced call to NC.SetupTitleBarMenu with calls to NC.InstallTitleBarButtonEventFn and NC.InstallTitleBarLeftMenu and
   ;; NC.InstallSketchTitleBarMiddleMenu.

   ;; fgh 6/30/86 Added NC.SK.COPY.BUTTONEVENTFN on Sketch Window. Added SKETCH.ADD.ELEMENT call when NoDisplayFlg

   ;; rht 6/11/88: Added RegionOrPosition arg and passed to NC.DetermineDisplayRegion. Also added ParamList and InterestedWindow args which
   ;; we ignore.

```
   (if NoDisplayFlg
       then (NC.SetSubstance Card (SKETCH.ADD.ELEMENT NIL NIL))
            Card
     else (LET (Window CardType)
               (SETQ CardType (NC.RetrieveType Card))
               [SETQ Window (SKETCHW.CREATE NIL NIL (NC.DetermineDisplayRegion Card RegionOrPosition)
                               (OR Title "Untitled")
                               NIL
                               (AND NC.AttachSketchMenu (SKETCH.COMMANDMENU (NC.GetCardTypeField
                                                                    MiddleButtonMenuItems
                                                                    CardType]
               (WINDOWPROP Window 'DONTQUERYCHANGES T)
               (WINDOWPROP Window 'COPYBUTTONEVENTFN (FUNCTION NC.SK.COPY.BUTTONEVENTFN))
               (WINDOWPROP Window 'SHRINKFN (LIST (FUNCTION NC.SketchCardShrinkFn)
                                                  (FUNCTION SHRINKATTACHEDWINDOWS)))
               (NC.InstallTitleBarButtonEventFn Window (FUNCTION NC.SketchTitleBarButtonEventFn))
               (NC.InstallTitleBarLeftMenu Window CardType)
               (NC.SetSubstance Card (INSURE.SKETCH Window))
               Window])
```

(**NC.BringUpSketchCard**
```
  [LAMBDA (Card Substance RegionOrPosition)                              ; Edited 14-Dec-88 13:40 by krivacic
                                                                         ; Bring up a sketch card containing substance in specified region
```
   ;; rht 2/1/85: Added windowprop to prevent sketch asking about saving changes.

   ;; rht 10/10/85: Now sets substance if was nil.

   ;; fgh 11/14/85 Updated to handle Card object.

   ;; rht 5/5/86: Replaced call to NC.SetupTitleBarMenu with calls to NC.InstallTitleBarButtonEventFn and NC.InstallTitleBarLeftMenu and
   ;; NC.InstallSketchTitleBarMiddleMenu.

   ;; fgh 6/30/86 Added NC.SK.COPY.BUTTONEVENTFN to sketch window. Made Substance to be INSURE.SKETCH of the sketch window.

   ;; rht 11/16/86: Now moves window to RegionOrPosition if already up.

   ;; rht 3/2/87: No longer passes RegionViewed and Scale to SKETCHW.CREATE. SKETCHW.CREATE will extract them from the imageobj.

   ;; pmi 2/18/88: Removed one of the two FLASHWs. Fixed up check for same position to be a little more efficient.

```
   (LET ([SketchName (AND (NULL Substance)
                          (MKATOM (fetch (Card UID) of Card]
        WindowRegion Title SketchViewer CardType)
        [COND
           [(SETQ SketchViewer (NC.FetchWindow Card))
            (TOTOPW SketchViewer)
            [if RegionOrPosition
                then (if (REGIONP RegionOrPosition)
                         then (if [NOT (EQUAL RegionOrPosition (WINDOWPROP SketchViewer 'REGION]
                                  then (SHAPEW SketchViewer (NC.DetermineDisplayRegion Card RegionOrPosition)))
                       elseif (POSITIONP RegionOrPosition)
                         then (SETQ WindowRegion (WINDOWPROP SketchViewer 'REGION))
                              (if [NOT (AND (EQUAL (fetch (POSITION XCOORD) of RegionOrPosition)
                                                  (fetch (REGION LEFT) of WindowRegion))
                                           (EQUAL (fetch (POSITION YCOORD) of RegionOrPosition)
```

```
                                               (fetch (REGION BOTTOM) of WindowRegion]
                                     then (SHAPEW SketchViewer (NC.DetermineDisplayRegion Card RegionOrPosition]
               (FLASHW SketchViewer)
               (TTY.PROCESS (WINDOWPROP SketchViewer 'PROCESS]
             (T (SETQ CardType (NC.RetrieveType Card))
                [SETQ SketchViewer (SKETCHW.CREATE (OR Substance SketchName)
                                                  NIL
                                                  (NC.DetermineDisplayRegion Card RegionOrPosition)
                                                  (NC.RetrieveTitle Card)
                                                  NIL
                                                  (SKETCH.COMMANDMENU (NC.GetCardTypeField MiddleButtonMenuItems CardType
                                                                        ]
               (WINDOWPROP SketchViewer 'DONTQUERYCHANGES T)
               (WINDOWPROP SketchViewer 'COPYBUTTONEVENTFN (FUNCTION NC.SK.COPY.BUTTONEVENTFN))
               (WINDOWPROP SketchViewer 'SHRINKFN (LIST (FUNCTION NC.SketchCardShrinkFn)
                                                        (FUNCTION SHRINKATTACHEDWINDOWS)))
               (NC.InstallTitleBarButtonEventFn SketchViewer (FUNCTION NC.SketchTitleBarButtonEventFn))
               (NC.InstallTitleBarLeftMenu SketchViewer CardType)
               (OR Substance (NC.SetSubstance Card (INSURE.SKETCH SketchViewer]
         SketchViewer])
```

(**NC.SketchDirtyP**
```
   [LAMBDA (Card)                                             (* rht%: " 2-Mar-87 21:10")
```

(* * rht 2/1/85%: No longer checks FILEPKGCHANGES. Now uses window prop.
Note that we really should have a flag on the sketch object itself.)

(* * fgh |11/14/85| Updated to handle Card object.)

(* * rht 3/2/87%: Took out calls to NC.FetchSavedRegion and NC.FetchScale.
Also fixed bug |#419:| sketch doing needless save after Put.)

```
   (EQ (WINDOWPROP (NC.FetchWindow Card)
            'SKETCHCHANGED)
       T])
```

(**NC.SketchTitleBarButtonEventFn**
```
   [LAMBDA (Window)                                           (* rht%: " 6-May-86 12:12")
```

(* * if inside title region and left button down, bringup left button title menu, otherwise just call the oldbuttoneventfn)

```
   (LET (LeftButtonMenu)
        (if (OR (INSIDEP (DSPCLIPPINGREGION NIL Window)
                         (LASTMOUSEX Window)
                         (LASTMOUSEY Window))
                (LASTMOUSESTATE MIDDLE))
            then (APPLY* (WINDOWPROP Window 'OLDBUTTONEVENTFN)
                         Window)
          elseif [type? MENU (SETQ LeftButtonMenu (WINDOWPROP Window 'TitleBarLeftButtonMenu]
            then (APPLY* (OR (MENU LeftButtonMenu)
                             (FUNCTION NILL))
                         Window])
```

(**NC.SketchCardCloseFn**
```
   [LAMBDA (CardIdentifier)                                   (* rht%: "16-Oct-86 17:01")
                                                              (* Quit from a sketch card, saving information on the database)
```

(* * rht 12/20/84%: Added ugly kludge, setting SKETCHCHANGED to nil, to keep sketch from asking whether to save
changes or not.)

(* * rht 2/1/85%: Removed above mentioned ugly kludge. It's now taken care of by a windowprop.
Now closes all sketch viewers for this sketch.)

(* * fgh |11/14/85| Updated to handle Card object.)

(* * rht 10/16/86%: Removed call to NC.DeactivateCard.)

```
   (LET (Card Window OldRegion NewRegion)
        (SETQ Card (NC.CoerceToCard CardIdentifier))
        (SETQ Window (NC.FetchWindow Card))
        [COND
           (Window (for Viewer in (ALL.SKETCH.VIEWERS (NC.FetchSubstance Card)) do (SKETCHW.CLOSEFN Viewer))
                   (SKED.CLEAR.SELECTION Window)
                   (for AttachedWindow in (ATTACHEDWINDOWS Window) do (DETACHWINDOW AttachedWindow)
                                                                      (CLOSEW AttachedWindow]
        (COND
           (Window (REMOVEPROMPTWINDOW Window)
                   (WINDOWPROP Window 'SKETCHOPMENU NIL)))
        (TTY.PROCESS T])
```

(**NC.SketchCardShrinkFn**
```
   [LAMBDA (W)                                                ; Edited  7-Dec-88 14:33 by krivacic
```

;; Check to make sure that icon's title agrees with card title. If not, retitle the icon. If this is first shrink then create a new icon and fill in title.

;; fgh 11/14/85 Updated to handle Card object.

;; rht 1/14/88: Now passes an icon position to SK.SHRINK.ICONCREATE if we had stashed one.

```
                                                    (* PROG ((OldIconTitle (WINDOWPROP W
                                                        (QUOTE SKETCH.ICON.TITLE)))
                                                        (Icon (WINDOWPROP W (QUOTE ICON)))
                                                        (Card (NC.CoerceToCard W)) IconTitle)
                                                    (COND ((NOT Icon) (SK.SHRINK.ICONCREATE W
                                                        (POSITIONP (WINDOWPROP W
                                                        (QUOTE SHRUNKENWINPOS))))
                                                        (SETQ Icon (WINDOWPROP W (QUOTE ICON)))))
                                                    (COND ((NOT (EQUAL OldIconTitle
                                                        (SETQ IconTitle (CONCAT "NC: "
                                                        (NC.RetrieveTitle Card)))) (WINDOWPROP W
                                                        (QUOTE SKETCH.ICON.TITLE) IconTitle)
                                                    (ICONTITLE IconTitle NIL NIL Icon))))
```

;; Generic Shrink Window

```
        (NC.ShrinkIcon W])
```

(**NC.SketchCopySubstance**
```
    [LAMBDA (Card FromStream ToStream Length)                          (* fgh%: "21-Nov-85 21:12")

            (* * Copy a sketch substance from FromStream to ToStream.)

            (* * fgh |11/14/85| Updated to handle Card object.)

            (* * fgh |11/20/85| NoteCards now handles the start and end ptrs --
            passing the from ptrs as args and properly setting the to ptrs.)

            (* * fgh |11/21/85| Now passes length instead of start and end ptrs.)

            (* * Copy the bytes)

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

(**NC.MarkSketchDirty**
```
    [LAMBDA (Card ResetFlg)                                            (* fgh%: "14-Nov-85 20:52")

            (* * Mark or unmark sketch as having been changed.)

            (* * rht 2/1/85%: Now goes through window prop rather than FILEPKG.)

            (* * fgh |11/14/85| Updated to handle Card object.)

        (WINDOWPROP (NC.FetchWindow Card)
                'SKETCHCHANGED ResetFlg])
```

(**NC.CollectReferencesInSketch**
```
    [LAMBDA (Card CheckAndDeleteFlg ReturnLinkIconsFlg ReturnLocationsFlg)
                                                                       (* rht%: "26-May-87 11:55")

            (* * Return a list of all links in sketch substance Substance. If CheckAndDeleteFlg, then delete any links found that are not
            valid links.)

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

            (* * fgh |11/14/85| Updated to handle Card object.)

            (* * rht 5/26/87%: Changed to match reduced functionality of NC.ValidLinkP, now have to check that destination of
            ActualLink is a valid card.)

        (LET ((SketchSubstance (NC.FetchSubstance Card))
            DirtyFlg)
            (CONS (for SketchElt in (SKETCH.LIST.OF.ELEMENTS SketchSubstance (FUNCTION NC.LinkIconSketchElementP))
                    bind LinkIcon CollectItem ActualLink when (PROGN [SETQ ActualLink (NC.FetchLinkFromLinkIcon
                                                                        (SETQ LinkIcon (
                                                                            SKETCH.IMAGEOBJ.OF.ELEMENT
                                                                                SketchElt]
                                                            (COND
                                                                ((NULL CheckAndDeleteFlg)
                                                            (* No checking required)
                                                                T)
                                                                ((AND (LISTP CheckAndDeleteFlg)
                                                                    (FMEMB (fetch (Link DestinationCard)
                                                                            of ActualLink)
                                                                        CheckAndDeleteFlg))
                                                            (* Already checked since ID cached on CheckAndDeleteFlg list)
                                                                T)
                                                                ((AND (NC.ValidLinkP ActualLink)
                                                                    (NC.ValidCardP (fetch (Link
```

```
                                                                              DestinationCard
                                                                                  )
                                                                           of ActualLink)))
                                                    (* Link is valid)
                                                        T)
                                                    (T
                                                    (* Link is bad. Replace it with the DeletedLink image object.)
                                                            (NC.DeleteLinkIconSketchElement
                                                                    SketchElt Card)
                                                        (SETQ DirtyFlg T)
                                                        NIL)))
                    collect (SETQ CollectItem (COND
                                                    (ReturnLinkIconsFlg LinkIcon)
                                                    (T ActualLink)))
                            (COND
                                (ReturnLocationsFlg (CONS CollectItem (SKETCH.POSITION.OF.ELEMENT SketchElt)))
                                (T CollectItem)))
                    DirtyFlg])
```

(**NC.GetSketchSubstance**

```
 [LAMBDA (Card Length Stream SubstanceVersion)                              ; Edited 14-Dec-88 16:28 by krivacic
```

    ;; Get sketch substance from Database stream. Database stream is positioned. READ the global sketch description, the locasl sketch scale and
    ;; region viewed. Also read in any cached bit maps for the MAPS system.

;;; fgh 11/14/85 Updated to handle Card and NoteFile objects.

;;; fgh 11/20/85 NoteCards now passes start and enptrs down.

;;; fgh 11/21/85 Now passed Length instead of start and end ptrs.

;;; rht 1/23/86: Now takes Stream as arg instead of computing from Card.

;;; rht 11/1/86: Now uses our readtable when reading.

;;; rht 2/28/87: Now uses new style of storing sketch on file whereby we package as imageobj.

;;; bk 12/14/88: Fix old styles to also return image objects.

```
        (DECLARE (GLOBALVARS NC.OrigReadTable))
```

;;; Get the substance

```
        (SELECTQ SubstanceVersion
            ((0 -1)
```

;;; Old style wrote down Scale and RegionViewed separately.

```
                (LET* ((Sketch (HREAD Stream))
                       (SKETCH-SCALE (READ Stream NC.OrigReadTable))
                       (SKETCH-REGION (READ Stream NC.OrigReadTable)))
                      (NC.SetScale Card SKETCH-SCALE)
                      (NC.SetRegionViewed Card SKETCH-REGION)
                      (while (EQ (READ Stream NC.OrigReadTable)
                                 '%###CACHEDMAP###)
                        do (NC.GetCachedMap Stream))
                      ;; Turn into a SKETCH IMAGEOBJ

                      (SKETCH.IMAGEOBJ (INSURE.SKETCH Sketch)
                             SKETCH-REGION SKETCH-SCALE)))
            (1
```

;;; New style expects sketch to be packaged as imageobj.

```
                (LET ((Sketch (HREAD Stream)))
                     (while (EQ (READ Stream NC.OrigReadTable)
                                '%###CACHEDMAP###)
                       do (NC.GetCachedMap Stream))
                     Sketch))
            NIL])
```

(**NC.PutSketchSubstance**

```
 [LAMBDA (Card Stream)                                                      ; Edited 14-Dec-88 16:39 by krivacic
```

    ;; Put the sketch substance for card ID to the database. Store the global sketch descriptor, the scale and region viewed for ID and any cached bit
    ;; maps.

;;; fgh 11/14/85 Updated to handle Card object.

;;; fgh 11/20/85 NoteCards now takes care of setting the start and end pointers for the substance.

;;; rht 1/23/86: Now takes Stream as arg instead of computing from Card.

;;; fgh 2/6/86 Now returns version number.

;;; fgh&rht 8/25/86 Changed call to NC.PutCachedMap to remove use of SCREENELT record.

;;; rht&pmi 10/15/86: Changed outdated SK.REGION.VIEWED call to SKETCH.REGION.VIEWED.

;;; rht 11/1/86: Now uses our readtable when printing.

;;; rht 2/28/87: Writes out sketches by first packaging as imageobject. This changes the format on the file, since we no longer need to write down region
;;; and scale separately.

```
(DECLARE (GLOBALVARS NC.OrigReadTable))
(LET ((Substance (NC.FetchSubstance Card))
      (Window (NC.FetchWindow Card)))
     (HPRINT (if (OPENWP Window)
                     then (MAKE.IMAGE.OBJECT.OF.SKETCH Window)
                   else Substance)
             Stream NIL T)
     (AND Window (for Element in (SKETCH.LIST.OF.ELEMENTS (INSURE.SKETCH Substance)
                                         [FUNCTION (LAMBDA (Element)
                                                     (EQ (GETSKETCHELEMENTPROP Element 'TYPE)
                                                         'MAP]
                                    T)
                     do (NC.PutCachedMap Element Window Stream)))
     (PRINT '%###ENDSKETCH### Stream NC.OrigReadTable)
     1])
```

(NC.PutCachedMap
```
[LAMBDA (SketchMapElement SketchWindow Stream)                    (* rht%: " 1-Nov-86 15:52")
                                                                 (* Put a cached bit map corresponding to MapScreenElt onto
                                                                 database file)
```

(* * |8/25/86| fgh&rht Removed use of SCREENELT record and replaced with call to CacheSpecsFromMapSketchElement
which is defined in the MAP package.)

(* * rht 11/1/86%: Now uses our readtable when printing.)

```
     (DECLARE (GLOBALVARS NC.OrigReadTable))
     (LET (CacheSpecs BitMap)
          (AND (LISTP (SETQ CacheSpecs (CacheSpecsFromMapSketchElement SketchMapElement SketchWindow)))
               (for CacheSpec in CacheSpecs do (BITMAPP (SETQ BitMap (APPLY (FUNCTION FetchCachedBitMap)
                                                                            CacheSpec)))
                               (PRINT '%###CACHEDMAP### Stream NC.OrigReadTable)
                               (PRINT CacheSpec Stream NC.OrigReadTable)
                               (HPRINT BitMap Stream T T]))
```

(NC.GetCachedMap
```
[LAMBDA (Stream)                                                 (* rht%: " 1-Nov-86 15:54")
```

(* Read a bit map from the file and then put it onto the cached maps list)

(* * FGH |9/28/85| Updated to handle new map changes. In particular, caching now done on the map resolution and noit on
the x and y offsets.)

(* * rht 10/10/86%: Replaced APPLY of SetCachedBitMap with NC.AutoloadApply so that NCMAPS will get autoloaded.)

(* * rht 11/1/86%: Now uses our readtable when reading.)

```
     (DECLARE (GLOBALVARS NC.OrigReadTable))
     (LET (CacheSpecs BitMap)
          (SETQ CacheSpecs (READ Stream NC.OrigReadTable))
          [COND
             ((EQ (LENGTH CacheSpecs)
                  8)
```

(* * Old Scheme for cahcing)

```
              (SETQ CacheSpecs (CONS 'HIGH (CDDR CacheSpecs]
          (SETQ BitMap (HREAD Stream))
          (AND CacheSpecs BitMap (NC.AutoloadApply (FUNCTION SetCachedBitMap)
                                          (CONS BitMap CacheSpecs])
```

(NC.UpdateLinkImagesInSketch
```
[LAMBDA (SourceCard DestinationCard)                             (* fgh%: "17-Nov-85 16:32")
                                                                (* For now do nothing since Link Images in Sketch have no
                                                                titles.)
```

(* * rht 8/20/85%: Rigged this to use Richard's new sketch programmer's interface.)

(* * fgh |11/17/85| Updated to handle card object.)

```
     (LET ((SketchWin (NC.FetchWindow SourceCard)))
          (for SketchElement in (SKETCH.LIST.OF.ELEMENTS SketchWin (FUNCTION NC.LinkIconSketchElementP))
             bind FoundAtLeastOneFlg when [NC.SameCardP DestinationCard (fetch (Link DestinationCard)
                                                                           of (NC.FetchLinkFromLinkIcon
                                                                                (SKETCH.IMAGEOBJ.OF.ELEMENT
                                                                                      SketchElement]
```

```
            do (SKETCH.ELEMENT.CHANGED SketchWin SketchElement)
               (SETQ FoundAtLeastOneFlg T)
         finally (COND
                    (FoundAtLeastOneFlg (NC.MarkCardDirty SourceCard))
```

(**NC.DelReferencesToCardFromSketch**
```
  [LAMBDA (SourceCard LinkOrDestinationCard Don'tCreateDeletedImageObjFlg)
                                                                (* rht%: " 4-Nov-86 14:52")
```

      (* * Remove all Link Icons pointing to Destination from the sketch Substance.)

      (* * rht 8/20/85%: Rewritten to use Richard's sketch programmer's interface.
Eliminates references to sketch records. I'm changing the innards of the affected link icons rather than deleting them and reinserting. That's because deleting the icon would cause a recursive call.)

      (* * kirk 14Nov85%: deleted use of LinkID)

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

```
    (PROG ((LinkFlg (type? Link LinkOrDestinationCard))
           DestinationCard SketchSubstance)
          (COND
             [LinkFlg (OR (NC.CardP SourceCard)
                         (SETQ SourceCard (fetch (Link SourceCard) of LinkOrDestinationCard]
             (T (SETQ DestinationCard LinkOrDestinationCard)))
          (SETQ SketchSubstance (NC.FetchSubstance SourceCard))
          (for SketchElement in (SKETCH.LIST.OF.ELEMENTS SketchSubstance (FUNCTION NC.LinkIconSketchElementP))
             bind LinkIcon when [PROGN (SETQ LinkIcon (SKETCH.IMAGEOBJ.OF.ELEMENT SketchElement))
                                       (COND
                                          (LinkFlg (NC.SameLinkP LinkOrDestinationCard (NC.FetchLinkFromLinkIcon
                                                                                          LinkIcon)))
                                          (T (NC.SameCardP DestinationCard (fetch (Link DestinationCard)
                                                                                of (NC.FetchLinkFromLinkIcon
                                                                                       LinkIcon]
             do (NC.DeleteLinkIconSketchElement SketchElement SourceCard Don'tCreateDeletedImageObjFlg])
```

(**NC.ExternalizeLinkIconsInSketch**
```
  [LAMBDA (Sketch)                                               (* rht%: "26-Mar-86 12:26")
```

      (* * Smashes all link icons in Sketch with external link icons.)

```
    [for SketchElement in (SKETCH.LIST.OF.ELEMENTS Sketch (FUNCTION NC.LinkIconSketchElementP)
                               T)
       do (LET* ((LinkIcon (SKETCH.IMAGEOBJ.OF.ELEMENT SketchElement))
                 (Link (NC.FetchLinkFromLinkIcon LinkIcon))
                 (NC.CoerceToExternalPutLinkIcon LinkIcon (fetch (Link SourceCard) of Link)
                        (fetch (Link DestinationCard) of Link)
                        (fetch (Link Label) of Link]
       Sketch])
```

(**NC.MakeExternalSketchCopy**
```
  [LAMBDA (SketchViewerOrImageObj)                               ; Edited 22-Jan-88 22:32 by Trigg
```

;;; Make a copy of the sketch smashing any link icons.

    ;; rht 4/22/87: Now installs proper grid, scale and region viewed in the copy. Now returns a sketch imageobj.

    ;; rht 1/22/88: Changed call of SKETCH.REGION.VIEWED to SKETCH.REGION.OF.SKETCH.

```
    (LET ((SketchCopy (SKETCH.ADD.ELEMENT NIL NIL)))
         (SKETCH.COPY.ELEMENTS (SKETCH.ELEMENTS.OF.SKETCH (INSURE.SKETCH SketchViewerOrImageObj))
                SketchCopy)
         (NC.ExternalizeLinkIconsInSketch SketchCopy)
         (MAKE.IMAGE.OBJECT.OF.SKETCH SketchCopy (SKETCH.REGION.OF.SKETCH SketchViewerOrImageObj)
                (SKETCH.VIEWER.SCALE SketchViewerOrImageObj)
                (SKETCH.VIEWER.GRID SketchViewerOrImageObj])
```

(**NC.SK.COPY.BUTTONEVENTFN**
```
  [LAMBDA (Window)                                               (* fgh%: "30-Jun-86 23:50")
```

      (* * Check to see if the TTY PROCESS is a Select NoteCards, if so then ask get user to choose a link icon and pass it back.
Otherwise, just call the normal sketch copybuttoneventfn.)

```
    (if (AND (WINDOWP (PROCESSPROP (TTY.PROCESS)
                        'WINDOW))
             (WINDOWPROP (PROCESSPROP (TTY.PROCESS)
                              'WINDOW)
                    'SelectingCards))
        then
```

      (* * Okay, TTY is a select NC process.)

```
           [LET [(ChosenElement (SKETCH.GET.ELEMENTS Window T (SKETCH.LIST.OF.ELEMENTS
                                                                  (INSURE.SKETCH Window)
                                                                  (FUNCTION (LAMBDA (Element)
```

```
                                                                (AND (EQ (GETSKETCHELEMENTPROP
                                                                          Element
                                                                          'TYPE)
                                                                         'SKIMAGEOBJ)
                                                                     (NC.LinkIconImageObjP
                                                                      (GETSKETCHELEMENTPROP
                                                                       Element
                                                                       'DATA]
                    (if ChosenElement
                        then (COPYINSERT (GETSKETCHELEMENTPROP ChosenElement 'DATA]
        else

            (* * TTY is not select NC, do what sketch does.)

            (SK.COPY.BUTTONEVENTFN Window])
)

[LOADINITADVISE NC-ADVICE0061 (CREATE.SKETCHW.COMMANDMENU (NIL (AFTER NIL (NC.RemoveSketchMenuItems !VALUE]
```

;;; This stuff supports the push-copy method of copying links in NC

```
(DEFINEQ
```

(**NC.TranslateWindowPositionToSketchPosition**
```
  [LAMBDA (Card Window WindowPositionX WindowPositionY)            (* fgh%: " 6-Feb-86 22:10")

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

     (create POSITION
             XCOORD _ WindowPositionX
             YCOORD _ WindowPositionY])
)
```

;;; Special stuff for autoloading maps

```
(DEFINEQ
```

(**NC.FakeMapInputFn**
```
  [LAMBDA (SketchWin)                                             (* rht%: " 9-Oct-86 14:55")

            (* * This just to force autoload of the NCMAPS library package.)

     (NC.AutoloadApply* (FUNCTION SK.MAP.INPUTFN)
            SketchWin])
```

(**NC.FakeMapInsideFn**
```
  [LAMBDA (Elem Region)                                           (* rht%: " 9-Oct-86 18:03")

            (* * This just to force autoload of the NCMAPS library package.)

     (NC.AutoloadApply* (FUNCTION SK.MAP.INSIDEFN)
            Elem Region])
)
```

```
(DECLARE%: DONTEVAL@LOAD

[IF NIL
    THEN ;; Don't load MAP stuff yet

            (NC.StoreAutoloadFnFile (FUNCTION SK.MAP.INPUTFN)
                    'NCMAPS
                    'NOTECARDSDIRECTORIES)
            (NC.StoreAutoloadFnFile (FUNCTION SK.MAP.INSIDEFN)
                    'NCMAPS
                    'NOTECARDSDIRECTORIES)
            (NC.StoreAutoloadFnFile (FUNCTION SetCachedBitMap)
                    'NCMAPS
                    'NOTECARDSDIRECTORIES)
            (OR (FMEMB 'MAP SKETCH.ELEMENT.TYPE.NAMES)
                (CREATE.SKETCH.ELEMENT.TYPE 'MAP "Map" "Forces autoload of MAP package." NIL NIL NIL NIL
                    (FUNCTION NC.FakeMapInputFn)
                    (FUNCTION NC.FakeMapInsideFn]
)
```

;;;; Add sketch card type to CardType list

```
(DEFINEQ
```

(**NC.AddSketchCard**
```
  [LAMBDA NIL                                                     ; Edited  3-Dec-87 19:01 by rht:
```

(* * fgh 11/14/85%: Updated to conform to merging of cardTypes and SubstanceTypes.)

(* * Added LinkIconAttachedBitMap field.)

(* * rht 2/28/87%: Ripped out old LAMBDA wrapper around GetFn.)

```
    (DECLARE (GLOBALVARS NC.SketchCardIcon))
    (NC.AddCardType 'Sketch 'NoteCard '[(MakeFn ,(FUNCTION NC.MakeSketchCard))
                                        (EditFn ,(FUNCTION NC.BringUpSketchCard))
                                        (QuitFn ,(FUNCTION NC.SketchCardCloseFn))
                                        (GetFn ,(FUNCTION NC.GetSketchSubstance))
                                        (PutFn ,(FUNCTION NC.PutSketchSubstance))
                                        (CopyFn ,(FUNCTION NC.SketchCopySubstance))
                                        (MarkDirtyFn ,(FUNCTION NC.MarkSketchDirty))
                                        (DirtyPFn ,(FUNCTION NC.SketchDirtyP))
                                        (CollectLinksFn ,(FUNCTION NC.CollectReferencesInSketch))
                                        (DeleteLinksFn ,(FUNCTION NC.DelReferencesToCardFromSketch))
                                        (UpdateLinkIconsFn ,(FUNCTION NC.UpdateLinkImagesInSketch))
                                        (InsertLinkFn ,(FUNCTION NC.InsertLinkInSketch))
                                        (TranslateWindowPositionFn ,(FUNCTION
                                                            NC.TranslateWindowPositionToSketchPosition)
                                            )
                                        (MiddleButtonMenuItems ,(SKETCH.COMMANDMENU.ITEMS NIL T]
            '((DefaultWidth 400)
              (DefaultHeight 350)
              (LinkAnchorModesSupported T)
              (LinkDisplayMode Title)
              (DisplayedInMenuFlg T)
              (LinkIconAttachedBitMap ,NC.SketchCardIcon])
)

(DECLARE%: DONTEVAL@LOAD
```

(**NC.AddSketchCard**)
)

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

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

FUNCTION INDEX

PROPERTY INDEX

VARIABLE INDEX