

File created: 3-Nov-2020 16:13:55 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCCROSSFILELINKS.;5

previous date: 9-Jan-94 19:46:07 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCCROSSFILELINKS.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1986, 1987, 1988, 1989, 1990, 1993, 1994, 2020 by Venue & Xerox Corporation. All rights reserved.

#### (RPAQQ NCCROSSFILELINKSCOMS

```
[(PROP (MAKEFILE-ENVIRONMENT FILETYPE)
  NCCROSSFILELINKS)
 (FNS NCAddStub.CrossFileLink)
```

;; This file defines a card type called CrossFileLink that is a stand-in for really being able to link across notefiles.

;; A programmer can customize the icon which will appear on the right side of the link icon for cross-file links. The main hook is the property  
;; 'AttachedBitMapFn' which should be placed on the card type's atom; which in this case is CrossFileLink. This function will be called in the  
;; display code by NC.FetchCrossFileLinkIconAttachedBitMap when a cross-file link is encountered. An example of such a function may be  
;; found in NC.CrossFileLinkAttachedBitMapFn in this file, and you would install it or your own function by doing (PUTPROP 'CrossFileLink  
;; 'AttachedBitMapFn (FUNCTION NC.CrossFileLinkAttachedBitMapFn)).

```
(GLOBALVARS NC.CopyCrossFileLinksMode NC.CrossFileLinkIcon NC.MakeCrossFileLinkIconStandIn
  NC.NewCrossFileLinksMode)
(INITVARS (NC.NewCrossFileLinksMode 'ASK))
(BITMAPS NC.CrossFileLinkIcon)
(INITVARS (NC.NewCrossFileLinksTwoWayFlg T)
  (NC.CopyCrossFileLinksMode NIL))
(FNS NC.AskCrossFileLinkMode NC.DeleteCrossFileLinkCard NC.FetchRemoteCrossFileLinkCard
  NC.CheckCrossFileLinkCardTitle NC.CheckCrossFileLinkType NC.MakeCrossFileLinkIconStandIn
  NC.ComputeCrossFileLinkMode)
(FNS NC.GetCrossFileLinkDestCard NC.CreateCrossFileLinkCard NC.CrossFileLinkCardP
  NC.OpenCrossFileLinkDestNoteFile)
(FNS NC.AddCrossFileLink NC.CrossFileLinkCopyFn NC.CrossFileLinkMakeFn NC.CrossFileLinkEditFn
  NC.CrossFileLinkGetFn NC.CrossFileLinkPutFn)
(DECLARE%: DONTVAL@LOAD (P (NC.AddCrossFileLink
  (PUTPROP 'CrossFileLink 'Don'tForceFilingFlg T))
```

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

```
(PUTPROPS NCCROSSFILELINKS FILETYPE :FAKE-COMPILE-FILE)
```

```
(DEFINEQ
```

#### (NCAddStub.CrossFileLink

```
[LAMBDA NIL
```

```
(* rht%: " 4-Oct-86 00:36")
```

```
(* * kirk 18Jun86 Add the CrossFileLink card stub)
```

```
(NC.AddCardTypeStub 'CrossFileLink 'NoteCard 'NCCROSSFILELINKS])
```

```
)
```

;; This file defines a card type called CrossFileLink that is a stand-in for really being able to link across notefiles.

;; A programmer can customize the icon which will appear on the right side of the link icon for cross-file links. The main hook is the property  
;; 'AttachedBitMapFn' which should be placed on the card type's atom; which in this case is CrossFileLink. This function will be called in the display  
;; code by NC.FetchCrossFileLinkIconAttachedBitMap when a cross-file link is encountered. An example of such a function may be found in  
;; NC.CrossFileLinkAttachedBitMapFn in this file, and you would install it or your own function by doing (PUTPROP 'CrossFileLink 'AttachedBitMapFn  
;; (FUNCTION NC.CrossFileLinkAttachedBitMapFn)).

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS NC.CopyCrossFileLinksMode NC.CrossFileLinkIcon NC.MakeCrossFileLinkIconStandIn
  NC.NewCrossFileLinksMode)
```

```
)
```

```
(RPAQQ? NC.NewCrossFileLinksMode 'ASK)
```

```
(RPAQQ? NC.CrossFileLinkIcon )
```

```
(RPAQQ? NC.NewCrossFileLinksTwoWayFlg T)
```

```
(RPAQQ? NC.CopyCrossFileLinksMode NIL)
```

```
(DEFINEQ
```

#### (NC.AskCrossFileLinkMode

```
[LAMBDA (DestCard InterestedWindow)
```

```
; Edited 10-Feb-88 11:25 by Trigg
```

:: Ask the user whether to make the link be twoway or oneway. Return T if user says two-way. If dest notefile is open read-only, then only choice is one-way so don't ask.

:: rht 2/10/88: Now checks whether prompt window was already open and if so, preserves that.

```
(LET* ((NoteFile (fetch (Card NoteFile) of DestCard))
      (FullFileName (fetch (NoteFile FullFileName) of NoteFile))
      (WasOpenPromptWindowFlg (NC.PromptWindowOpenP InterestedWindow)))
  (if (NC.ReadOnlyNoteFileP NoteFile)
      then NIL
      else (NC.AskYesOrNo (CONCAT "Okay to make cross-file link to '" (NC.RetrieveTitle DestCard)
                                "' in '"
                                (FILENAMEFIELD FullFileName 'NAME)
                                "';"
                                (FILENAMEFIELD FullFileName 'VERSION)
                                "' be TWOWAY?"
                                (CHARACTER 13)
                                " (otherwise leaves no record of link in destination notefile) ")
                            " -- " "Yes" T InterestedWindow WasOpenPromptWindowFlg]))
```

**(NC.DeleteCrossFileLinkCard**

[LAMBDA (CrossFileLinkCard) (\* rht%: "27-May-87 21:54")

(\* Delete the crossfile link card. If it's a twoway crossfile link, and its "twin" is in an open notefile, then delete the cross file link over there.)

```
(LET ((DestCard (NC.FetchRemoteCrossFileLinkCard CrossFileLinkCard)))
  (if DestCard
      then (NC.DeleteNoteCardInternal DestCard T)
      (NC.DeleteNoteCardInternal CrossFileLinkCard T))
```

**(NC.FetchRemoteCrossFileLinkCard**

[LAMBDA (CrossFileLinkCard) (\* rht%: "27-May-87 21:54")

(\* Return the remote "twin" of CrossFileLinkCard if its notefile is open.)

```
(LET ((CrossFileLinkSubstance (NCP.CardSubstance CrossFileLinkCard))
      DestCard RemoteCrossFileLinkCardUID NoteFileUID NoteFile)
  (if [AND (fetch (CrossFileLinkSubstance CrossFileLinkTwoWayFlg) of CrossFileLinkSubstance)
        (SETQ RemoteCrossFileLinkCardUID (fetch (CrossFileLinkSubstance RemoteCrossFileLinkCardUID)
                                                of CrossFileLinkSubstance))
        (type? UID (SETQ NoteFileUID (fetch (CrossFileLinkSubstance CrossFileLinkDestNoteFileUID)
                                             of CrossFileLinkSubstance)))
        (NCP.OpenNoteFileP (SETQ NoteFile (NC.NoteFileFromNoteFileUID NoteFileUID NoteFileUID)))
        (NC.ValidCardP (SETQ DestCard (NC.CardFromUID RemoteCrossFileLinkCardUID NoteFile)))]
      then DestCard
      else NIL))
```

**(NC.CheckCrossFileLinkCardTitle**

[LAMBDA (CrossFileLinkCard DestinationCard) (\* rht%: "27-May-87 21:47")

(\* If titles don't agree, change crossfilelink card title to match.)

```
(LET ((DestinationCardTitle (NC.RetrieveTitle DestinationCard))
      (if (NOT (STREQUAL DestinationCardTitle (NC.RetrieveTitle CrossFileLinkCard)))
          then (NC.AssignTitle CrossFileLinkCard NIL DestinationCardTitle))
```

**(NC.CheckCrossFileLinkType**

[LAMBDA (RemoteCrossFileLinkCard LocalCrossFileLinkCard Link) (\* rht%: "27-May-87 22:33")

(\* If link types don't agree, change remote crossfilelink type to match.)

```
(LET ((RemoteDestCard (NC.GetCrossFileLinkDestCard LocalCrossFileLinkCard NIL T))
      (LinkLabel (fetch (Link Label) of Link))
      RemoteLink NoteFile)
  (if RemoteDestCard
      then [SETQ RemoteLink (OR (for FromLink in (NC.RetrieveFromLinks RemoteCrossFileLinkCard)
                               when (NC.SameCardP (fetch (Link SourceCard) of FromLink)
                                                    RemoteDestCard)
                               do (RETURN FromLink))
                               (for ToLink in (NC.RetrieveToLinks RemoteCrossFileLinkCard)
                               when (NC.SameCardP (fetch (Link DestinationCard) of ToLink)
                                                    RemoteDestCard)
                               do (RETURN ToLink))]
          (if (NOT (EQ (fetch (Link Label) of RemoteLink)
                      LinkLabel))
              then (if [NOT (NCP.ValidLinkTypeP LinkLabel (SETQ NoteFile (fetch (Card NoteFile)
                                                                                  of RemoteCrossFileLinkCard))
                    then (NCP.CreateLinkType LinkLabel NoteFile))
                  (NC.RelabelLink RemoteLink NIL LinkLabel))]
```

**(NC.MakeCrossFileLinkIconStandIn**

[LAMBDA (CrossFileLinkCard) (\* rht%: "27-May-87 23:39")

(\* Return an imageobj to act as a standin for a cross file link.)

```
(DECLARE (GLOBALVARS NC.ExternalPutLinkIconImageFns))
(IMAGEOBJCREATE (CONCAT "[[ Cross-file link to '" (NC.RetrieveTitle CrossFileLinkCard)
                  "' ]]" )
                NC.ExternalPutLinkIconImageFns])
```

### (NC.ComputeCrossFileLinkMode

```
[LAMBDA (RemoteCard CrossFileLinkModePropList InterestedWindow)
        (* rht%: " 6-Jun-87 16:19")
```

(\* If we've already made cross file links to RemoteCard's notefile, then it'll be registered on CrossFileLinkModePropList. Otherwise, consult global var, possibly ask user, and register her answer on CrossFileLinkModePropList.)

(\* rht 6/6/87%: Changed so that check of globalvar happens before check of CrossFileLinkModePropList.)

```
(DECLARE (GLOBALVARS NC.NewCrossFileLinksMode))
(LET ((NoteFile (fetch (Card NoteFile) of RemoteCard))
      Mode)
      (COND
        ((FMEMB NC.NewCrossFileLinksMode ' (TWOWAY ONEWAY))
         NC.NewCrossFileLinksMode)
        ((FMEMB NoteFile CrossFileLinkModePropList)
         (LISTGET CrossFileLinkModePropList NoteFile))
        (T (SETQ Mode (if (NC.AskCrossFileLinkMode RemoteCard InterestedWindow)
                          then 'TWOWAY
                          else 'ONEWAY))
           (LISTPUT CrossFileLinkModePropList NoteFile Mode)
           Mode]))
```

)

(DEFINEQ

### (NC.GetCrossFileLinkDestCard

```
[LAMBDA (CrossFileLinkCard InterestedWindow Don'tOpenDestNoteFileFlg)
        ; Edited 4-Oct-88 18:35 by DSJ
```

(\* Find the notefile corresponding to this crossfilelink and try to open it if not already open. Then look for the card in there having the given UID. Return NIL if failed for any reason.)

(\* rht 11/10/86%: Make sure CrossFileLinkCard is cached before fetching substance.)

(\* rht 11/19/86%: Now rips off version number from destination notefile hint.)

(\* rht 12/11/86%: Now checks that destination card is not deleted.)

(\* rht 12/16/86%: Now takes InterestedWindow argument.)

(\* rht 3/25/87%: Now calls NC.CoerceToInterestedWindow.)

(\* rht 3/27/87%: Now calls NC.OpenCrossFileLinkDestNoteFile.)

(\* rht 5/27/87%: New arg Don'tOpenDestNoteFileFlg. If non-nil, then destination notefile has to be already open.)

(\* rht 6/4/87%: Now passes CrossFileLinkCard to NC.OpenCrossFileLinkDestNoteFile.)

(\* rht 6/8/87%: Now computes WasOpenPromptWindowFlg)

(\* pmi 8/7/87%: Now gets a filename if it has to loop and ask for another notefile to try.)

(\* pmi 9/8/87%: Now sets Don'tOpenDestNoteFileFlg to the results of NC.OpenCrossFileLinkDestNoteFile)

(\* dsj |9/14/88.| For speed, Now no longer updates the CrossFileLinkDestFileHint field of the CrossFileLinkCard substance with the DestNoteFile, thereby avoiding a call to FULLNAME.)

(\* dsj |10/4/88.| Changed change of |9/14/88.| Now still does replaces but does not call FULLNAME.)

```
(LET ((Title (NC.FetchTitle CrossFileLinkCard))
      (SourceNoteFile (fetch (Card NoteFile) of CrossFileLinkCard))
      Substance DestNoteFileUID DestFileHint DestCardUID DestNoteFile DestFileName Card
      WasOpenPromptWindowFlg)
      (OR InterestedWindow (SETQ InterestedWindow (NC.CoerceToInterestedWindow SourceNoteFile)))
      (SETQ WasOpenPromptWindowFlg (NC.PromptWindowOpenP InterestedWindow))
      (if (NOT (NC.ActiveCardP CrossFileLinkCard))
          then (NC.GetNoteCard CrossFileLinkCard))
      (SETQ Substance (NC.FetchSubstance CrossFileLinkCard))
      (SETQ DestCardUID (fetch (CrossFileLinkSubstance CrossFileLinkDestCardUID) of Substance))
      (SETQ DestNoteFileUID (fetch (CrossFileLinkSubstance CrossFileLinkDestNoteFileUID) of Substance))
      (SETQ DestFileHint (fetch (CrossFileLinkSubstance CrossFileLinkDestFileHint) of Substance))
      (SETQ DestNoteFile (NC.NoteFileFromNoteFileUID DestNoteFileUID))
      (SETQ DestFileName (OR (AND (type? NoteFile DestNoteFile)
                                  (fetch (NoteFile FullFileName) of DestNoteFile))
                             DestFileHint))
      (for while [if Don'tOpenDestNoteFileFlg
```

```

    then (NCP.OpenNoteFileP DestNoteFile)
    else [SETQ Don'tOpenDestNoteFileFlg (NULL (SETQ DestNoteFile (NC.OpenCrossFileLinkDestNoteFile
                                                DestNoteFile DestFileName
                                                InterestedWindow
                                                CrossFileLinkCard]
                                                (SETQ DestFileName (OR DestFileName (AND (type? NoteFile DestNoteFile)
                                                                 (fetch (NoteFile FullFileName) of DestNoteFile))
bind NewFileName
do (if (NC.ValidCardP (SETQ Card (NC.CardFromUID DestCardUID DestNoteFile)))
    then
        ;; Now does not call FULLNAME on the CrossFileLinkDestFileHint, but instead strips the version off the FullFileName.
        ;; dsj.
        (if (NOT (EQUAL (PACKFILENAME 'VERSION NIL 'BODY (SETQ NewFileName (fetch (NoteFile
                                                                                               FullFileName)
                                                                                               of DestNoteFile)))
                        (fetch (CrossFileLinkSubstance CrossFileLinkDestFileHint) of Substance)))
            then (replace (CrossFileLinkSubstance CrossFileLinkDestFileHint) of Substance
                          with (PACKFILENAME 'VERSION NIL 'BODY NewFileName))
            (NC.MarkCardDirty CrossFileLinkCard))
        (RETURN Card)
    else (if Don'tOpenDestNoteFileFlg
        then (RETURN NIL)
        else (NC.PrintMsg InterestedWindow T "Couldn't find destination card in " DestFileName
            "." (CHARACTER 13))
            (if (NOT (NC.AskYesOrNo "Want to try opening a different notefile? " " -- " "Yes" NIL
                InterestedWindow WasOpenPromptWindowFlg))
                then (RETURN NIL)
                else (SETQ DestNoteFile (SETQ DestFileName NIL]))

```

**(NC.CreateCrossFileLinkCard**

```

[LAMBDA (SourceCard DestCard TwoWayFlg) (* rht%: " 9-Dec-86 14:50")

    (** Create a cross file link card in SourceCard's notefile.)

    (** rht 11/19/86%: Now rips off version number from destination notefile hint.)

    (** rht 12/9/86%: No longer records fact that card was "just created.")

(LET ((DestNoteFile (fetch (Card NoteFile) of DestCard))
      CrossFileLinkCard)
  (SETQ CrossFileLinkCard (NC.MakeNoteCard 'CrossFileLink (fetch (Card NoteFile) of SourceCard)
      (NC.FetchTitle DestCard)
      T
      (create CrossFileLinkSubstance
              CrossFileLinkDestCardUID _ (fetch (Card UID) of DestCard)
              CrossFileLinkDestNoteFileUID _ (fetch (NoteFile UID) of DestNoteFile)
              CrossFileLinkDestFileHint _ (PACKFILENAME 'VERSION NIL
              'BODY
              (fetch (NoteFile FullFileName) of DestNoteFile))
              CrossFileLinkTwoWayFlg _ TwoWayFlg)))
      (* Indicate that card was just created.))
  CrossFileLinkCard])

```

**(NC.CrossFileLinkCardP**

```

[LAMBDA (Card) (* rht%: " 6-Oct-86 10:49")

    (** Return non-nil if card is a crossfilelink card.)

(EQ (NC.FetchType Card)
    'CrossFileLink])

```

**(NC.OpenCrossFileLinkDestNoteFile**

```

[LAMBDA (DestNoteFile DestFileName InterestedWindow CrossFileLinkCard)
; Edited 11-Jan-89 10:29 by rtk

```

;;; If DestNoteFile is an open notefile, then fine. Otherwise get file names from user and keep trying to open until she gives up.  
 ;;; rht 5/25/87: Minor change: no longer passes InterestedWindow to NC.OpenNoteFile.  
 ;;; rht 6/3/87: Now calls new function NC.AskUserWithMenu. Added CrossFileLinkCard argument whose title is used to construct Message.  
 ;;; rht 6/8/87: Now computes WasOpenPromptWindowFlg  
 ;;; pmi 8/6/87: Replaced NC.AskUser with NC.DatabaseFileName when asking the user for a filename to try. This insures that the filename ends in  
 ;;; '.notefile'  
 ;;; bk 1/10/89: Set the Don'tCreate flag when calling NC.OpenNoteFile

```

(OR InterestedWindow (NC.CoerceToInterestedWindow CrossFileLinkCard))
(LET ((WasOpenPromptWindowFlg (NC.PromptWindowOpenP InterestedWindow)))

```

```

(if (NCP.OpenNoteFileP DestNoteFile)
  then DestNoteFile
  else (LET [(MenuItems ' (Read/Write Read-Only Don't% Open)
             (for while (OR DestFileName (SETQ DestFileName (NC.DatabaseFileName "File name to try
                                                                opening: " NIL T T NIL InterestedWindow
                                                                )))
             do (LET ((Message (if (NC.ValidCardP CrossFileLinkCard)
                                   then (CONCAT "Open " DestFileName " to look for '" (
                                                                                          NC.RetrieveTitle
                                                                                          CrossFileLinkCard
                                                                                          )
                                   else (CONCAT "OPEN " DestFileName "?"))
             ReadOnlyOpenFlg)
            (if [AND (SELECTQ (NC.AskUserWithMenu MenuItems Message InterestedWindow
                                                WasOpenPromptWindowFlg T)
                    (Read/Write (SETQ ReadOnlyOpenFlg NIL)
                                T)
                    (Read-Only (SETQ ReadOnlyOpenFlg T)
                                (Don't% Open (RETURN NIL))
                                (RETURN NIL))
                    (NCP.OpenNoteFileP (SETQ DestNoteFile
                                            (NC.OpenNoteFile (OR DestNoteFile DestFileName)
                                                            NIL T NIL NIL NIL NIL NIL NIL NIL NIL
                                                            ReadOnlyOpenFlg)
                                then (RETURN DestNoteFile)
                                else (NC.PrintMsg InterestedWindow T "Couldn't open " DestFileName "."
                                     (CHARACTER 13))
                                     (SETQ DestFileName (SETQ DestNoteFile NIL))
                                     (if (NOT (NC.AskYesOrNo "Want to try opening a different notefile? " " -- "
                                                             "Yes" NIL InterestedWindow WasOpenPromptWindowFlg))
                                         then (RETURN NIL)]))
  )
)

```

(DEFINEQ

**(NC.AddCrossFileLink**

[LAMBDA NIL

; Edited 3-Dec-87 19:01 by rht:

(\* rht 11/19/86%: Added LinkIconAttachedBitMap field.)

```

(DECLARE (GLOBALVARS NC.CrossFileLinkIcon))
(NC.AddCardType 'CrossFileLink 'NoteCard `[MakeFn ,(FUNCTION NC.CrossFileLinkMakeFn))
          (EditFn ,(FUNCTION NC.CrossFileLinkEditFn))
          (GetFn ,(FUNCTION NC.CrossFileLinkGetFn))
          (PutFn ,(FUNCTION NC.CrossFileLinkPutFn))
          (CopyFn ,(FUNCTION NC.CrossFileLinkCopyFn))
          (MarkDirtyFn ,(FUNCTION NIL))
          (DirtyPFn ,(FUNCTION NIL))
          (QuitFn ,(FUNCTION NIL))
          (DeleteLinksFn ,(FUNCTION NIL))
          (InsertLinkFn ,(FUNCTION NIL))
          (CollectLinksFn ,(FUNCTION NIL))
          (UpdateLinkIconsFn ,(FUNCTION NIL))
          (TranslateWindowPositionFn ,(FUNCTION NIL))
  `((LinkAnchorModesSupported NIL)
    [LinkDisplayMode ,(NC.InsureLinkDisplayMode 'Title)
     LinkIconAttachedBitMap ,NC.CrossFileLinkIcon])
)

```

**(NC.CrossFileLinkCopyFn**

[LAMBDA (Card FromStream ToStream Length)

(\* rht%: " 3-Oct-86 20:20")

(\* Copy a CrossFileLink substance from FromStream to ToStream.)

```

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

```

**(NC.CrossFileLinkMakeFn**

[LAMBDA (Card Title NoDisplayFlg CrossFileLinkSubstance)

(\* rht%: " 3-Oct-86 20:42")

(\* Make a CrossFileLink substance.)

```

(NC.SetSubstance Card CrossFileLinkSubstance)
Card])

```

**(NC.CrossFileLinkEditFn**

[LAMBDA (Card Substance ReadOnly RegionOrPosition TypeSpecificArgs)

(\* Randy.Gobbel " 4-Nov-87 14:12")

(\* Given a CrossFileLink Substance, try to find the corresponding destination card in the destination notefile and bring it up.)

(\* rht 11/13/86%: Now updates title of crossfile link card to be the same as destination card if necessary.)

(\* rht 5/27/87%: Now calls NC.CheckCrossFileLinkCardTitle.)

(\* rg |11/4/87| added ReadOnly)

```
(LET ((DestinationCard (NC.GetCrossFileLinkDestCard Card))
      DestinationCardTitle)
  (if DestinationCard
      then (NC.CheckCrossFileLinkCardTitle Card DestinationCard)
           (NC.EditNoteCard DestinationCard (OR ReadOnly (fetch (NoteFile ReadOnlyFlg)
                                                                of (fetch (Card NoteFile) of Card)))
                RegionOrPosition TypeSpecificArgs))
```

**(NC.CrossFileLinkGetFn**

[LAMBDA (Card Length Stream VersionNum) (\* rht%: "27-May-87 15:06")

(\* Get the crossfilelink substance from the disk)

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

(\* rht 5/27/87%: Now reads in new format of CrossFileLinkSubstance record with two new fields.)

```
(DECLARE (GLOBALVARS NC.OrigReadTable))
(if (LEQ VersionNum 0)
    then (create CrossFileLinkSubstance
                 CrossFileLinkDestCardUID _ (NC.ReadUID Stream)
                 CrossFileLinkDestNoteFileUID _ (NC.ReadUID Stream)
                 CrossFileLinkDestFileHint _ (READ Stream NC.OrigReadTable))
    else (create CrossFileLinkSubstance
                 CrossFileLinkDestCardUID _ (NC.ReadUID Stream)
                 CrossFileLinkDestNoteFileUID _ (NC.ReadUID Stream)
                 RemoteCrossFileLinkCardUID _ (NC.ReadUID Stream)
                 CrossFileLinkDestFileHint _ (READ Stream NC.OrigReadTable)
                 CrossFileLinkTwoWayFlg _ (READ Stream NC.OrigReadTable))
```

**(NC.CrossFileLinkPutFn**

[LAMBDA (Card Stream) (\* rht%: "27-May-87 15:07")

(\* Put crossfilelink substance)

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

(\* rht 5/27/87%: Changed to write down new fields RemoteCrossFileLinkCardUID and CrossFileLinkTwoWayFlg of CrossFileLinkSubstance record.)

```
(DECLARE (GLOBALVARS NC.OrigReadTable))
(LET ((Substance (NC.FetchSubstance Card)))
  (NC.WriteUID Stream (fetch (CrossFileLinkSubstance CrossFileLinkDestCardUID) of Substance))
  (NC.WriteUID Stream (fetch (CrossFileLinkSubstance CrossFileLinkDestNoteFileUID) of Substance))
  (NC.WriteUID Stream (fetch (CrossFileLinkSubstance RemoteCrossFileLinkCardUID) of Substance))
  (PRINT (fetch (CrossFileLinkSubstance CrossFileLinkDestFileHint) of Substance)
         Stream NC.OrigReadTable)
  (PRINT (fetch (CrossFileLinkSubstance CrossFileLinkTwoWayFlg) of Substance)
         Stream NC.OrigReadTable))
```

1])

)

(DECLARE%: DONTEVAL@LOAD

**(NC.AddCrossFileLink**

(PUTPROP 'CrossFileLink 'Don'tForceFilingFlg T)

)

(PUTPROPS NCCROSSFILELINKS COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1988 1989 1990 1993 1994 2020))

---

**FUNCTION INDEX**

NC.AddCrossFileLink .....	5	NC.CrossFileLinkCardP .....	4	NC.DeleteCrossFileLinkCard .....	2
NC.AskCrossFileLinkMode .....	1	NC.CrossFileLinkCopyFn .....	5	NC.FetchRemoteCrossFileLinkCard ..	2
NC.CheckCrossFileLinkCardTitle ...	2	NC.CrossFileLinkEditFn .....	5	NC.GetCrossFileLinkDestCard .....	3
NC.CheckCrossFileLinkType .....	2	NC.CrossFileLinkGetFn .....	6	NC.MakeCrossFileLinkIconStandIn ..	2
NC.ComputeCrossFileLinkMode .....	3	NC.CrossFileLinkMakeFn .....	5	NC.OpenCrossFileLinkDestNoteFile ..	4
NC.CreateCrossFileLinkCard .....	4	NC.CrossFileLinkPutFn .....	6	NCAddStub.CrossFileLink .....	1

---

**VARIABLE INDEX**

NC.CopyCrossFileLinksMode .....	1	NC.NewCrossFileLinksMode .....	1
NC.CrossFileLinkIcon .....	1	NC.NewCrossFileLinksTwoWayFlg .....	1

---

**PROPERTY INDEX**

NCCROSSFILELINKS .....	1
------------------------	---

---