

File created: 10-Mar-2024 23:35:48 {DSK}<home>frank<il>notecards>library>NCHACKS.;3

changes to: (FNS NCHACKS.RemoveDeletedIconsFromTextCard NCHACKS.ReorderLinkIconsInTextCard)

previous date: 8-Mar-2024 12:42:57 {DSK}<home>frank<il>notecards>library>NCHACKS.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **NCHACKSCOMS**

(

;;; global text search and global text replace.

(FNS NCHACKS.GlobalTextReplace NCHACKS.TextSearch NCHACKS.GlobalTextReplaceInCard
NCHACKS.TextSearchInCard)

;;; remove deleted icons from text cards.

(FNS NCHACKS.RemoveDeletedIconsFromTextCards NCHACKS.RemoveDeletedIconsFromTextCard)

;;; sort link icons in text cards.

(FNS NCHACKS.ReorderLinkIconsInTextCards NCHACKS.ReorderLinkIconsInTextCard)

;;; search for card parts with dates in a specified range.

(FNS NCHACKS.DateSearch NCHACKS.DateSearchInCard)

;;; search for cards which have been modified/created within given time periods.

(FNS NCHACKS.CardsModifiedBetweenDates)

;;; make a chain linking cards using a given link type.

(FNS NCHACKS.MakeChain)

;;; Make link icons invisible in text cards.

(FNS NCHACKS.MakeLinkIconsInvisible NCHACKS.MakeLinkIconsInvisibleInTextCard
NCHACKS.MakeLinkIconForLinkInvisible NCHACKS.MakeTEditCharInvisible)
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
NCHACKS))

;;; global text search and global text replace.

(DEFINEQ

(NCHACKS.GlobalTextReplace

[LAMBDA (String1 String2 WildCards? CardsOrNoteFile)

(* rht%: "28-Apr-87 11:12")

(* CardsOrNoteFile should be either a single card, a list of cards, or an open notefile, in which case all the cards in it are checked. Only cards of types inheriting from Text are considered.

This goes through the Cards list replacing every occurrence of text string String1 by String2.

WildCards? non-nil means wild card characters may appear in String1.

matches any single character, * matches any sequence of characters, and %' can be used to quote one of the wildcard characters.)

(* rht 4/28/87%: Took out second arg to FUNCTION.)

(if (NCP.OpenNoteFileP CardsOrNoteFile)

then [NCP.MapCards CardsOrNoteFile (FUNCTION (LAMBDA (Card)

(NCHACKS.GlobalTextReplaceInCard String1 String2 WildCards?
Card)

else (for Card in (MKLIST CardsOrNoteFile) when (NCP.ValidCardP Card)

do (NCHACKS.GlobalTextReplaceInCard String1 String2 WildCards? Card])

(NCHACKS.TextSearch

[LAMBDA (String WildCards? CardsOrNoteFile ReturnLocsFlg)

(* rht%: "28-Apr-87 11:16")

(* CardsOrNoteFile should be either a single card, a list of cards, or an open notefile, in which case all the cards in it are checked. Only cards of types inheriting from Text are considered.

This goes through the Cards list looking for occurrences of text string String.

WildCards? non-nil means wild card characters may appear in String.

matches any single character, * matches any sequence of characters, and %' can be used to quote one of the wildcard characters. If ReturnLocsFlg is nil, then just return list of cards containing at least one occurrence of String.

If ReturnLocsFlg is non-nil, then return a list of lists. Each list has first element a card id followed by the locations of

occurrences of String. These are single integers if WildCards? is nil.
If WildCards? is non-nil, then these are lists of start and end locations of the matching string.)

(* rht 4/28/87%: Changed APPLY of NCONC to call to MAPCONC.)

```
(if (NCP.OpenNoteFileP CardsOrNoteFile)
  then [MAPCONC (NCP.MapCards CardsOrNoteFile [FUNCTION (LAMBDA (Card)
    (NCHACKS.TextSearchInCard String WildCards? Card
      ReturnLocsFlg]
      (FUNCTION TRUE))
    (FUNCTION (LAMBDA (X)
      X])]
  else (for Card in (MKLIST CardsOrNoteFile) when (NCP.ValidCardP Card)
    join (NCHACKS.TextSearchInCard String WildCards? Card ReturnLocsFlg])
```

(NCHACKS.GlobalTextReplacelnCard

[LAMBDA (String1 String2 WildCards? Card) (* rht%: "17-Jul-86 18:01")

(* Replaces every occurrence of text string String1 by String2 in contents of Card.
WildCards? non-nil means wild card characters may appear in String1.
%# matches any single character, * matches any sequence of characters, and QUOTE can be used to quote one of the
wildcard characters.)

```
(if (NCP.TextBasedP Card)
  then (LET ((WasActiveFlg (NCP.CardCachedP Card))
    (Length (NCHARS String1))
    (Loc 0)
    TextStream)
    (if (NOT WasActiveFlg)
      then (NCP.CacheCards Card))
    (SETQ TextStream (NCP.CardSubstance Card))
    (while (SETQ Loc (TEDIT.FIND TextStream String1 (ADD1 Loc)
      NIL WildCards?))
      do (if WildCards?
        then [SETQ Length (ADD1 (DIFFERENCE (CADR Loc)
          (CAR Loc))
          (SETQ Loc (CAR Loc)))
          (TEDIT.DELETE TextStream Loc Length)
          (TEDIT.INSERT TextStream String2 Loc))
        (if (NOT WasActiveFlg)
          then (NCP.UncacheCards Card]))
```

(NCHACKS.TextSearchInCard

[LAMBDA (String WildCards? Card ReturnLocsFlg) (* rht%: " 5-May-87 17:22")

(* This goes through Card looking for occurrences of text string String.
WildCards? non-nil means wild card characters may appear in String.
%# matches any single character, * matches any sequence of characters, and %' can be used to quote one of the wildcard
characters. If no occurrence found, then return NIL. Otherwise, if ReturnLocsFlg is nil, then return a list of this card.
Else return a list having as first element a card id followed by the locations of occurrences of String.
These are single integers if WildCards? is nil. If WildCards? is non-nil, then these are lists of start and end locations of the
matching string.)

(* rht 5/5/87%: Now passes non-nil %#START arg to TEDIT.FIND when ReturnLocsFlg is nil.)

```
(if (NCP.TextBasedP Card)
  then (LET ((WasActiveFlg (NCP.CardCachedP Card))
    (Length (NCHARS String))
    (Loc 0)
    Hits TextStream)
    (if (NOT WasActiveFlg)
      then (NCP.CacheCards Card))
    (SETQ TextStream (NCP.CardSubstance Card))
    (PROG1 (if ReturnLocsFlg
      then (if [SETQ Hits (while (SETQ Loc (TEDIT.FIND TextStream String (ADD1 Loc)
        NIL WildCards?))
        collect (PROG1 Loc
          (if WildCards?
            then (SETQ Loc (CAR Loc)))))]
        then (LIST (CONS Card Hits)))
      else (if (TEDIT.FIND TextStream String 1 NIL WildCards?)
        then (LIST Card)))
      (if (NOT WasActiveFlg)
        then (NCP.UncacheCards Card))))))
```

;;; remove deleted icons from text cards.

(DEFINEQ

(NCHACKS.RemoveDeletedIconsFromTextCards

[LAMBDA (CardsOrNoteFile) (* rht%: "17-Jul-86 18:13")

(* For cards in Cards of substance type TEXT, remove all deleted link icons from their substance.)

```
(if (NCP.OpenNoteFileP CardsOrNoteFile)
  then [NCP.MapCards CardsOrNoteFile (FUNCTION (LAMBDA (Card)
    (NCHACKS.RemoveDeletedIconsFromTextCard Card)
  )
)
else (for Card in (MKLIST CardsOrNoteFile) when (NCP.ValidCardP Card) do (
  NCHACKS.RemoveDeletedIconsFromTextCard
  Card))
```

(NCHACKS.RemoveDeletedIconsFromTextCard

[LAMBDA (Card)

(* rht%: "18-Jul-86 14:34")

(* Remove all deleted link icons from Card's substance.)

```
(if (NCP.TextBasedP Card)
  then (LET ((WasActiveFlg (NCP.CardCachedP Card))
    TextStream)
    (if (NOT WasActiveFlg)
      then (NCP.CacheCards Card))
    (SETQ TextStream (NCP.CardSubstance Card)) (* Need to reverse list so that we delete the last icon first.)
    (for IconPair in (REVERSE (NC.TEditCollectObjects TextStream (FUNCTION NC.DeletedLinkImageObjP)
      )
    )
      do (TEDIT.DELETE TextStream (CADR IconPair)
        1))
    (if (NOT WasActiveFlg)
      then (NCP.UncacheCards Card))
  )
)
```

;;; sort link icons in text cards.

(DEFINEQ

(NCHACKS.ReorderLinkIconsInTextCards

[LAMBDA (CardsOrNoteFile OrderingFn QuietFlg)

(* rht%: "28-Apr-87 11:13")

(* Resort the link icons occuring in Cards according to OrderingFn.
If OrderingFn is the litatom ALPHABETIZE, then use NC.IDAlphOrder.
If OrderingFn is NIL, then use the OrderingFn prop of the card or NC.IDAlphOrder if none.
Print message to prompt window unless QuietFlg is non-nil.)

(* rht 4/28/87%: Took out second arg to FUNCTION.)

```
(if (EQ OrderingFn 'ALPHABETIZE)
  then (SETQ OrderingFn (FUNCTION NC.IDAlphOrder)))
(if (NCP.OpenNoteFileP CardsOrNoteFile)
  then [NCP.MapCards CardsOrNoteFile (FUNCTION (LAMBDA (Card)
    (NCHACKS.ReorderLinkIconsInTextCard Card OrderingFn QuietFlg
  )
)
else (for Card in (MKLIST CardsOrNoteFile) when (NCP.ValidCardP Card) do (NCHACKS.ReorderLinkIconsInTextCard
  Card OrderingFn QuietFlg))
(OR QuietFlg (NCP.PrintMsg NIL T "Done."))
```

(NCHACKS.ReorderLinkIconsInTextCard

[LAMBDA (Card OrderingFn QuietFlg)

(* rht%: "18-Jul-86 15:34")

(* Resort the link icons occuring in Card according to OrderingFn.
If the latter is NIL, then use the OrderingFn prop of the card, or NC.IDAlphOrder if none such.)

```
(if (NCP.TextBasedP Card)
  then (OR QuietFlg (NCP.PrintMsg NIL T "Sorting link icons in " Card " ..."))
  (LET ((WasActiveFlg (NCP.CardCachedP Card))
    TextStream OrderingFnToUse LinkIconsAndLocs SortedCardsAndLocs)
    (if (NOT WasActiveFlg)
      then (NCP.CacheCards Card))
    [if (SETQ OrderingFnToUse (OR OrderingFn (NCP.CardProp Card 'OrderingFn)
      (FUNCTION NC.IDAlphOrder)))
      then (SETQ TextStream (NCP.CardSubstance Card))
      (SETQ LinkIconsAndLocs (NC.TEditCollectObjects TextStream (FUNCTION
        NC.LinkIconImageObjP)))
      [SETQ SortedCardsAndLocs (SORT (for LinkIconAndLoc in LinkIconsAndLocs as i
        from 0 eachtime (BLOCK)
        collect (LIST (NC.FetchLinkFromLinkIcon (CAR
          LinkIconAndLoc
        )
          (CADR LinkIconAndLoc)))
        (FUNCTION (LAMBDA (CardAndLoc1 CardAndLoc2)
          (APPLY* OrderingFnToUse (CAR CardAndLoc1)
            (CAR CardAndLoc2)
          )
        )
      )
      (for RestOfSortedCardsAndLocs on SortedCardsAndLocs as LinkIconAndLoc in
        LinkIconsAndLocs
        eachtime (BLOCK) bind Loc FromLoc ToLoc Sel1 Sel2
        do
```

(* Location to move FROM is taken from next pair in sorted list.)

```
(TEDIT.SETSEL TextStream (SETQ FromLoc (CADAR RestOfSortedCardsAndLocs))
  1
  'LEFT)
(SETQ Sel1 (TEDIT.GETSEL TextStream))
```

(* Location to move TO is just next pair on unsorted list, except that there may be intervening guys that will later move out of the way.)

```
(TEDIT.SETSEL TextStream (SETQ ToLoc (PLUS (SETQ Loc (CADR LinkIconAndLoc))
  (for CardAndLoc
    in (CDR RestOfSortedCardsAndLocs)
    eachtime (BLOCK)
    when (LESSP (CADR CardAndLoc)
      Loc)
    count CardAndLoc)))
  0
  'LEFT)
(SETQ Sel2 (TEDIT.GETSEL TextStream))
(TEDIT.MOVE Sel1 Sel2)
```

(* Now increment any locs of pairs whose location has changed by virtue of this move.)

```
(for CardAndLoc in (CDR RestOfSortedCardsAndLocs) bind Loc eachtime (BLOCK)
  when (AND (LESSP (SETQ Loc (CADR CardAndLoc))
    FromLoc)
    (LEQ ToLoc Loc))
  do (RPLACA (CDR CardAndLoc)
    (ADD1 Loc)
    (if (NOT WasActiveFlg)
      then (NCP.UncacheCards Card)))
```

::: search for card parts with dates in a specified range.

(DEFINEQ

(NCHACKS.DateSearch

```
[LAMBDA (DateString1 DateString2 CardsOrNoteFile) (* rht%: "28-Apr-87 11:15")
```

(* There should be an open notefile. This goes through the Cards list looking for occurrences of card parts modified between the dates DateString1 and DateString2. If DateString1 is NIL, then defaults to a very early date. If DateString2 is NIL, then defaults to current date.)

(* rht 4/28/87%: Changed APPLY of NCONC to call to MAPCONC.)

```
(LET [(IDate1 (OR (IDATE (OR DateString1 " 1-Jan-70 00:00:00"))
  (IDATE " 1-Jan-70 00:00:00")))
  (IDate2 (OR (IDATE (OR DateString2 (DATE)))
  (IDATE (DATE)))]
  (if (NCP.OpenNoteFileP CardsOrNoteFile)
    then [MAPCONC (NCP.MapCards CardsOrNoteFile [FUNCTION (LAMBDA (Card)
      (NCHACKS.DateSearchInCard IDate1 IDate2 Card)
      ]
      (FUNCTION TRUE))
      (FUNCTION (LAMBDA (X)
        X))]
    else (for Card in (MKLIST CardsOrNoteFile) when (NCP.ValidCardP Card)
      join (NCHACKS.DateSearchInCard IDate1 IDate2 Card))])
```

(NCHACKS.DateSearchInCard

```
[LAMBDA (IDate1 IDate2 Card) (* rht%: "17-Jul-86 17:56")
```

(* This goes through Card's card parts looking for those modified between the numeric dates IDate1 and IDate2.)

```
(LET ((WasActiveFlg (NCP.CardCachedP Card))
  Hits IDate Date Dates)
  (if (NOT WasActiveFlg)
    then (NCP.CacheCards Card))
  (SETQ Dates (NCP.GetCardDates Card))
  (SETQ Hits (LDIFFERENCE (LIST (if (AND [LESSP IDate1 (SETQ IDate (IDATE (SETQ Date (fetch (NOTECARDDATES
    SUBSTANCEDATE)
    of Dates]
    (LESSP IDate IDate2))
    then (LIST 'SUBSTANCEDATE Date))
    (if (AND [LESSP IDate1 (SETQ IDate (IDATE (fetch (NOTECARDDATES LINKSDATE)
    of Dates]
    (LESSP IDate IDate2))
    then (LIST 'LINKSDATE Date))
    (if (AND [LESSP IDate1 (SETQ IDate (IDATE (fetch (NOTECARDDATES TITLEDATE)
    of Dates]
    (LESSP IDate IDate2))
    then (LIST 'TITLEDATE Date))
```

```

        (if (AND [LESSP IDate1 (SETQ IDate (IDATE (fetch (NOTECARDDATES
                                                    PROPLISTDATE)
                                                    of Dates]
                                                    (LESSP IDate IDate2))
            then (LIST 'PROPLISTDATE Date)))
            (LIST NIL)))
    (PROG1 (if Hits
            then (LIST (CONS Card Hits))
            else NIL)
        (if (NOT WasActiveFlg)
            then (NCP.UncacheCards Card))))
)

```

;;; search for cards which have been modified/created within given time periods.

(DEFINEQ

(NCHACKS.CardsModifiedBetweenDates

```

[LAMBDA (CardsOrNoteFile CardTypes ModifiedRange LastModifiedRange CreatedRange)
    (* pmi%: "18-Aug-87 11:14")

```

(* pmi 7/14/87%: First created to answer the question%: Which cards in CardsOrNoteFile of card type (one of CardTypes) were modified or last modified or created between EarlierDate and LaterDate? If ModifiedOrLastModifiedOrCreated is NIL, (QUOTE Modified) is used. If EarlierDate is NIL, it succeeds on any date before LaterDate. If LaterDate is NIL, it succeeds on any date after EarlierDate. If CardTypes is NIL, all card types are searched. CardsOrNoteFile should be a card, a list of cards, or an open notefile.)

(* 8/18/87%: Modified to take three date ranges, one for modified, one for last modified, and one for created. Each range should be a list of start and end dates of the form DD-Mon-YY, where Mon contains the first three letters of the month. Returns the intersection of cards which were modified/created between these three date ranges.)

```

(LET (EarlierModDate LaterModDate EarlierLastModDate LaterLastModDate EarlierCreateDate LaterCreateDate)
    (if (LISTP ModifiedRange)
        then (SETQ EarlierModDate (OR (IDATE (CONCAT (CAR ModifiedRange)
                                                    " 00:00:00"))
            0))
        (SETQ LaterModDate (OR (IDATE (CONCAT (CADR ModifiedRange)
                                                    " 23:59:59"))
            MAX.INTEGER)))
    (if (LISTP LastModifiedRange)
        then (SETQ EarlierLastModDate (OR (IDATE (CONCAT (CAR LastModifiedRange)
                                                    " 00:00:00"))
            0))
        (SETQ LaterLastModDate (OR (IDATE (CONCAT (CADR LastModifiedRange)
                                                    " 23:59:59"))
            MAX.INTEGER)))
    (if (LISTP CreatedRange)
        then (SETQ EarlierCreateDate (OR (IDATE (CONCAT (CAR CreatedRange)
                                                    " 00:00:00"))
            0))
        (SETQ LaterCreateDate (OR (IDATE (CONCAT (CADR CreatedRange)
                                                    " 23:59:59"))
            MAX.INTEGER)))
    (for Card in (NCP.CardsOfTypes CardsOrNoteFile CardTypes) bind CardUpdates CardDate
        when (AND (SETQ CardUpdates (MAPCAR (NCP.CardProp Card 'Updates)
            'CADR))
            (if (LISTP ModifiedRange)
                then (for Date in CardUpdates when (AND (SETQ CardDate (IDATE Date))
                    (IGEQ CardDate EarlierModDate)
                    (ILEQ CardDate LaterModDate))
                do (RETURN T) finally (RETURN NIL))
            else T)
            (if (LISTP LastModifiedRange)
                then (SETQ CardDate (IDATE (CAR CardUpdates)))
                    (if (AND (IGEQ CardDate EarlierLastModDate)
                        (ILEQ CardDate LaterLastModDate))
                        then T
                        else NIL)
            else T)
            (if (LISTP CreatedRange)
                then [SETQ CardDate (IDATE (CAR (LAST CardUpdates]
                    (if (AND (IGEQ CardDate EarlierCreateDate)
                        (ILEQ CardDate LaterCreateDate))
                        then T
                        else NIL)
            else T))
        collect Card])
)

```

;;; make a chain linking cards using a given link type.

(DEFINEQ

(NCHACKS.MakeChain

[LAMBDA (LinkType Cards Position AddCRFlg) (* rht%: "17-Jul-86 18:05")

(* * Create links between successive cards in Cards each of type LinkType positioned at Position.)

(for RestOfCards on Cards bind (LastCard _ (CAR (LAST Cards)))
Card

until (EQ (SETQ Card (CAR RestOfCards))
LastCard)

when (AND (NCP.ValidCardP Card)
(NCP.TextBasedP Card))

do (if (AND AddCRFlg (EQ Position 'END))
then (NCP.CardAddText Card (CHARACTER 13)
Position))

(NCP.CreateLink (LIST Card Position)
(CADR RestOfCards)
LinkType)

(if (AND AddCRFlg (EQ Position 'START))
then (NCP.CardAddText Card (CHARACTER 13]))

)

::: Make link icons invisible in text cards.

(DEFINEQ

(NCHACKS.MakeLinkIconsInvisible

[LAMBDA (CardsOrLinksOrNoteFile Invisibility) (* rht%: "28-Apr-87 11:16")

(* * CardsOrLinksOrNoteFile should be one of%: a card, a link, a list of cards and links, or a notefile.
Make any link icons for given links or contained in given cards or notefile that are in text cards invisible.
Invisibility should be one of the atoms ON or OFF.)

(* rht 4/28/87%: Took out second arg to FUNCTION.)

(if (NCP.OpenNoteFileP CardsOrLinksOrNoteFile)
then [NCP.MapCards CardsOrNoteFile (FUNCTION (LAMBDA (Card)
(NCHACKS.MakeLinkIconsInvisibleInTextCard Card Invisibility)]

else (for CardOrLink in (MKLIST CardsOrLinksOrNoteFile) do (COND
(NCP.ValidCardP CardOrLink)
(NCHACKS.MakeLinkIconsInvisibleInTextCard
CardOrLink Invisibility))
(NCP.ValidLinkP CardOrLink)
(NCHACKS.MakeLinkIconForLinkInvisible CardOrLink
Invisibility]))

(NCHACKS.MakeLinkIconsInvisibleInTextCard

[LAMBDA (Card Invisibility) (* rht%: "18-Jul-86 21:54")

(* * Make any link icons appearing in Card invisible or visible according to whether Invisibility is ON or OFF.)

(if (NCP.TextBasedP Card)
then (LET ((WasActiveFlg (NCP.CardCachedP Card))
TextStream)
(if (NOT WasActiveFlg)
then (NCP.CacheCards Card))
(SETQ TextStream (NCP.CardSubstance Card))
(for LinkPair in (CAR (NCP.ApplyCardTypeFn CollectLinksFn Card NIL T T))
do (NCHACKS.MakeTEditCharInvisible TextStream (CDR LinkPair)
Invisibility))
(if (NOT WasActiveFlg)
then (NCP.UncacheCards Card]))

(NCHACKS.MakeLinkIconForLinkInvisible

[LAMBDA (Link Invisibility) (* rht%: "18-Jul-86 21:54")

(* * Make any link icons appearing in Card invisible or visible according to whether Invisibility is ON or OFF.)

(LET ((Card (NCP.LinkSource Link)))
(if (NCP.TextBasedP Card)
then (LET ((WasActiveFlg (NCP.CardCachedP Card))
TextStream)
(if (NOT WasActiveFlg)
then (NCP.CacheCards Card))
(SETQ TextStream (NCP.CardSubstance Card))
(for LinkPair in (CAR (NCP.ApplyCardTypeFn CollectLinksFn Card NIL NIL T))
when (NC.SameLinkP Link (CAR LinkPair)) do (NCHACKS.MakeTEditCharInvisible TextStream
(CDR LinkPair)
Invisibility))
(if (NOT WasActiveFlg)
then (NCP.UncacheCards Card]))

(NCHACKS.MakeTEditCharInvisible

[LAMBDA (TextStream Loc Invisibility)

; Edited 4-Dec-87 10:26 by rht:

(* * Make whatever char is at Loc be invisible or visible according to whether Invisibility is ON or OFF.)

(TEDIT.LOOKS TextStream `(INVISIBLE ,Invisibility)
(TEDIT.SETSEL TextStream Loc 1 'LEFT])

)

(PUTPROPS **NCHACKS FILETYPE** :TCOMPL)

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

FUNCTION INDEX

NCHACKS.CardsModifiedBetweenDates	5	NCHACKS.MakeLinkIconsInvisibleInTextCard	6
NCHACKS.DateSearch	4	NCHACKS.MakeTEditCharInvisible	7
NCHACKS.DateSearchInCard	4	NCHACKS.RemoveDeletedIconsFromTextCard	3
NCHACKS.GlobalTextReplace	1	NCHACKS.RemoveDeletedIconsFromTextCards	2
NCHACKS.GlobalTextReplaceInCard	2	NCHACKS.ReorderLinkIconsInTextCard	3
NCHACKS.MakeChain	6	NCHACKS.ReorderLinkIconsInTextCards	3
NCHACKS.MakeLinkIconForLinkInvisible	6	NCHACKS.TextSearch	1
NCHACKS.MakeLinkIconsInvisible	6	NCHACKS.TextSearchInCard	2

PROPERTY INDEX

NCHACKS	7
---------------	---
