

File created: 26-Jul-88 14:25:20 {QV}<NOTECARDS>1.3L>LIBRARY>NCHISTORYCARD.;3

changes to: (FNS NCHist.HistoryCardMakeFn NCHist.NoteFileWhenOpenedFn NCHist.ComputeHistory  
NCHist.TurnLoggingOn NCHist.TurnLoggingOff NCHist.AddHistoryCard)  
(VARS NCHISTORYCARDCOMS)

previous date: 20-Jun-88 17:47:16 {QV}<NOTECARDS>1.3L>LIBRARY>NCHISTORYCARD.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1986, 1987, 1988 by Xerox Corporation. All rights reserved.

#### (RPAQQ NCHISTORYCARDCOMS

```
((DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT NCHISTORYCARD))  
[DECLARE%: COPY FIRST (P (NC.LoadFileFromDirectories 'NCCOLLABORATORCARD)  
(NC.LoadFileFromDirectories 'NCLOGGER]
```

#### ;;; Package of History Card and handy functions

```
(FNS NCHist.HistoricizeNoteFile NCHist.NoteFileWhenOpenedFn NCHist.CreateHistoryCardTitle  
NCHist.GetHistoryFileBox)  
(FNS NCHist.HistoryCardMakeFn NCHist.HistoryCardWhenSavedFn NCHist.AddHistoryCard)  
(FNS NCHist.ShaveLeadingSpace)
```

#### ;;; General automated history card functions

```
(FNS NCHist.CreateHistoryDate NCHist.ComputeHistory NCHist.GetLogEventFromStream NCHist.ProcessLogEvent  
NCHist.ChangeHistoryEventSpecs NCHist.AskHistoryEventSpecs  
NCHist.InitializeHistoryEventsSpecsStylesheet NCHist.InitializeHistoryEventHashArray  
NCHist.InitializeLogEventFnHashArray NCHist.InitializeEventCategories NCHist.ConvertPointersToLinks  
NCHist.StartLogging NCHist.SuspendLogging NCHist.StopLogging NCHist.StartNoteFileLogging  
NCHist.SuspendNoteFileLogging NCHist.StartHistoryCardLogging NCHist.SuspendHistoryCardLogging)
```

#### ;;; Fns for the various event category functions

```
(FNS NCHist.CreateLinkEventsMenuItemsFn NCHist.CreateLinkEventsDefaultMenuChoicesFn  
NCHist.CreateLinkEventsMenuChoicesFn NCHist.DeleteLinkEventsMenuItemsFn  
NCHist.DeleteLinkEventsDefaultMenuChoicesFn NCHist.DeleteLinkEventsMenuChoicesFn)
```

#### ;;; Fns called to display the events in the History Card

```
(FNS NCHist.EditCard.BeginDisplayFn NCHist.EditCard.EndDisplayFn NCHist.QuitCard.BeginDisplayFn  
NCHist.QuitCard.EndDisplayFn NCHist.MakeCard.BeginDisplayFn NCHist.MakeCard.EndDisplayFn  
NCHist.GetNewCardDisplayFn NCHist.DelCardDisplayFn NCHist.CacheCardDisplayFn  
NCHist.UncacheCardDisplayFn NCHist.SaveCard.BeginDisplayFn NCHist.SaveCard.EndDisplayFn  
NCHist.PutLinksDisplayFn NCHist.PutMainCardDataDisplayFn NCHist.PutPropListDisplayFn  
NCHist.PutTitleDisplayFn NCHist.AssignTitle.BeginDisplayFn NCHist.AssignTitle.EndDisplayFn  
NCHist.MakeLinkDisplayFn NCHist.DelLinkDisplayFn NCHist.RecomputeBrowser.BeginDisplayFn  
NCHist.RecomputeBrowser.EndDisplayFn NCHist.RelayoutBrowser.BeginDisplayFn  
NCHist.RelayoutBrowser.EndDisplayFn NCHist.AddCardTypeDisplayFn NCHist.StartLoggingDisplayFn)  
(RECORDS EventCategory)
```

#### ;;; Global variables settable by the user

```
(GLOBALVARS NCHist.HistoryCardPosition NCHist.InhibitInitialLogging)  
(INITVARS (NCHist.HistoryCardPosition NIL)  
(NCHist.InhibitInitialLogging NIL))
```

#### ;;; Other global variables

```
(GLOBALVARS NCHist.HistoryEventsSpecsStylesheet NCHist.HistoryEventHashArray NCHist.LogEventFnHashArray  
NCHist.HistoryLinkType NCHist.EventCategories NCHist.CardEvents NCHist.CreateLinkEvents  
NCHist.DeleteLinkEvents NCHist.BrowserEvents NCHist.CardTypeEvents)  
(VARS (NCHist.HistoryLinkType 'History)  
(NCHist.HistoryEventHashArray (HASHARRAY NCLOG.HashArraySize))  
(NCHist.LogEventFnHashArray (HASHARRAY NCLOG.HashArraySize))  
(NCHist.CardEvents (create EventCategory Name _ 'Card% Events MenuItems _  
'(Create% Card |Bring Up Card| Close% Card |Change Card Substance|  
|Change Card Prop List| |Change Card Title| Delete% Card)  
MenuItemsFn _ NIL DefaultMenuItems _ '(Create% Card  
|Change Card Substance|  
|Change Card Title|  
Delete% Card)  
DefaultMenuItemsFn _ NIL MenuChoicesFn _ NIL))  
(NCHist.CreateLinkEvents (create EventCategory Name _ '|Create Link Events| MenuItems _ NIL
```

```

MenuItemsFn _ (FUNCTION NCHist.CreateLinkEventsMenuItemsFn)
DefaultMenuItems _ NIL DefaultMenuItemsFn _ (FUNCTION
    NCHist.CreateLinkEventsDefaultMenuChoicesFn
)
MenuChoicesFn _ (FUNCTION NCHist.CreateLinkEventsMenuChoicesFn))
(NCHist.DeleteLinkEvents (create EventCategory Name _ '|Delete Link Events| MenuItems _ NIL
    MenuItemsFn _ (FUNCTION NCHist.DeleteLinkEventsMenuItemsFn)
    DefaultMenuItems _ NIL DefaultMenuItemsFn _ (FUNCTION
        NCHist.DeleteLinkEventsDefaultMenuChoicesFn
    )
    MenuChoicesFn _ (FUNCTION NCHist.DeleteLinkEventsMenuChoicesFn))
(NCHist.BrowserEvents (create EventCategory Name _ 'Browser% Events MenuItems _
    '(Recompute% Browser Relayout% Browser)
    MenuItemsFn _ NIL DefaultMenuItems _ '(Recompute% Browser)
    DefaultMenuItemsFn _ NIL MenuChoicesFn _ NIL))
(NCHist.CardTypeEvents (create EventCategory Name _ '|Card Type Events| MenuItems _ '(
    |Add Card Type|
    )
    MenuItemsFn _ NIL DefaultMenuItems _ NIL DefaultMenuItemsFn _ NIL
    MenuChoicesFn _ NIL))
(P (NCHist.AddHistoryCard)
(PUTPROP 'History 'WhenSavedFn (FUNCTION NCHist.HistoryCardWhenSavedFn))
(NCHist.InitializeHistoryEventsSpecsStylesheet)
(NCHist.InitializeHistoryEventHashArray)
(NCHist.InitializeLogEventFnHashArray)
(NCHist.InitializeEventCategories))))

```

(DECLARE%: DONTCOPY

```

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

```

(DECLARE%: COPY FIRST

(NC.LoadFileFromDirectories 'NCCOLLABORATORCARD)

```

(NC.LoadFileFromDirectories 'NCLOGGER)
)

```

;;; Package of History Card and handy functions

(DEFINEQ

(NCHist.HistoricizeNoteFile

[LAMBDA (NoteFile)

; Edited 20-Jun-88 17:22 by pmi

```

;;; Initialize given notefile for History Card use. Make a History Use Filebox if none exists and register in the notefile. Hang a call to
;;; NCHist.NoteFileWhenOpenedFn in the OpenEventsCard for the notefile.

```

;;; rht 4/19/87: Now calls NC.LoadFileFromDirectories rather than FILESLOAD.

```

(if (NCP.OpenNoteFileP NoteFile)
  then (LET ([OpenEventSExp '(PROGN (NC.LoadFileFromDirectories 'NCHISTORYCARD)
    (NCHist.NoteFileWhenOpenedFn NoteFile)
    (OpenEventsCard (NCP.GetOpenEventsCard NoteFile))
    OpenEventsCardWasCachedFlg)

```

;;; Add call to NCHist.NoteFileWhenOpenedFn to the open events card substance.

```

(if (NOT (SETQ OpenEventsCardWasCachedFlg (NCP.ActiveCardP OpenEventsCard)))
  then (NCP.CacheCards OpenEventsCard))
[if (NOT (MEMBER OpenEventSExp (NCP.CardSubstance OpenEventsCard)))
  then (NCP.CardSubstance OpenEventsCard (CONS OpenEventSExp (NCP.CardSubstance
    OpenEventsCard)
    (if (NOT OpenEventsCardWasCachedFlg)
      then (NCP.CloseCards OpenEventsCard))

```

;;; Run the history OpenNoteFileFn now.

```

(NCHist.NoteFileWhenOpenedFn NoteFile))
else (NCP.ReportError "NCHist.HistoritizeNoteFile" (CONCAT "Can't historitize closed notefile: " NoteFile))

```

(NCHist.NoteFileWhenOpenedFn

[LAMBDA (NoteFile)

; Edited 26-Jul-88 10:35 by pmi

;; Function to be called when we open a NoteFile that has been 'Historitized.' Creates and opens a new History card for this session

;; rht 2/4/87: Now calls NCP.CreateCard with NoDisplayFlg NIL.

;; pmi 10/22/87: Now places new History card at (0, 0) if NCHist.HistoryCardPosition is not set.

;; pmi 7/26/88: Adds NoteFile Icon middle button items for starting and suspending logging on this notefile.

```

(DECLARE (GLOBALVARS NCHist.HistoryCardPosition))
(LET (HistoryCard)
  (NCHist.GetHistoryFileBox NoteFile)
  [NCP.OpenCard (SETQ HistoryCard (NCP.CreateCard 'History NoteFile NIL T))
    (OR NCHist.HistoryCardPosition (SETQ NCHist.HistoryCardPosition

```

```

                (create POSITION
                        XCOORD _ 0
                        YCOORD _ 0]
(NCP.AddNoteFileIconMiddleButtonItems NoteFile '((Start% Logging NCHist.StartNoteFileLogging "Start
logging for all active History cards in this
notefile." (SUBITEMS (Suspend% Logging
NCHist.SuspendNoteFileLogging
"Suspend logging
for all active
History cards in
this notefile."))

```

**(NCHist.CreateHistoryCardTitle**

[LAMBDA (Card)

; Edited 20-Jun-88 17:22 by pmi

;;; Creates title for history card consisting of date and initials.

```

(CONCAT (NCHist.CreateHistoryDate)
" "
(NCP.NoteFileProp (NCP.CardNoteFile Card)
'Initials])

```

**(NCHist.GetHistoryFileBox**

[LAMBDA (NoteFile)

; Edited 20-Jun-88 17:25 by pmi

;;; Create a history filebox if there isn't one already.

```

(LET ((HistoryFileBox (NCP.LookupCardByName 'HistoryFileBox NoteFile)))
  (if (NOT (NCP.ValidCardP HistoryFileBox))
    then (SETQ HistoryFileBox (NCP.CreateFileBox NoteFile "NoteFile Use History" T NIL NIL (
NCP.ContentsFileBox
NoteFile)))
    (NCP.RegisterCardByName 'HistoryFileBox HistoryFileBox))
(NCP.AddSpecialCard HistoryFileBox)
HistoryFileBox])

```

)

(DEFINEQ

**(NCHist.HistoryCardMakeFn**

[LAMBDA (Card Title NoDisplayFlg)

; Edited 26-Jul-88 10:40 by pmi

```

;; pmi 10/29/87: Now turns on logging.
;; pmi 7/12/88: Now stores the NCLogInfo on the notefile, instead of on the history card.
;; pmi 7/26/88: Checks NCHist.InhibitInitialLogging before turning logging on.
(DECLARE (GLOBALVARS NCHist.EventCategories NCHist.InhibitInitialLogging))
(LET ((NoteFile (NCP.CardNoteFile Card))
      ReturnValue HistoryFileBox DateString)
  ;; If one doesn't exist, create a NoteFile use history filebox, otherwise find the old one.
  (SETQ HistoryFileBox (NCHist.GetHistoryFileBox NoteFile))
  ;; Create a text card, then give it a title and file it
  (SETQ ReturnValue (NCP.ApplySuperTypeFn MakeFn Card Title NoDisplayFlg))
  ;; Cache the history filebox so that NCP.FileCards won't have to
  ;; close it.
  (OR (NCP.CardCachedP HistoryFileBox)
      (NCP.CacheCards HistoryFileBox))
  (NCP.FileCards Card HistoryFileBox)
  (NCP.CardTitle Card (OR Title (NCHist.CreateHistoryCardTitle Card)))
  (NCP.NoteFileAddProp NoteFile 'HistoryCards Card)
  ;; Put the start time in the beginning of the card's substance in bold face.
  (SETQ DateString (CONCAT (DATE (DATEFORMAT NO.DATE NO.SECONDS))
" - "))
  (NCP.CardAddText Card (CONCAT DateString (CHARACTER 13))
' START)
  (TEDIT.LOOKS (NCP.CardSubstance Card)
' (WEIGHT BOLD)
1
(NCHARS DateString))
  (if NCHist.InhibitInitialLogging
    else (NCHist.StartLogging Card))
  ReturnValue])

```

; Turn on logging for automated History cards.

**(NCHist.HistoryCardWhenSavedFn**

[LAMBDA (HistoryCard)

; Edited 20-Jun-88 17:26 by pmi

```

;;; QuitFn for the History card type. Delete old close time from the History card and insert the current close time in its place. Only do this for 'current'
;;; history cards, i.e. ones opened during this session.

```

```

(if (for Card in (NCP.NoteFileProp (NCP.CardNoteFile HistoryCard)

```

```

        'HistoryCards)
    when (NCP.ValidCardP Card) thereis (NCP.SameCardP Card HistoryCard))
then (LET ((Stream (NCP.CardSubstance HistoryCard))
    Time AfterDashPosition CRPosition)
    [SETQ Time (CONCAT " " (DATE (DATEFORMAT NO.DATE NO.SECONDS]
    (SETQ AfterDashPosition (ADD1 (TEDIT.FIND Stream "-" 1)))
    (SETQ CRPosition (TEDIT.FIND Stream (CHARACTER 13)
    1))
    (TEDIT.DELETE Stream AfterDashPosition (IDIFFERENCE CRPosition AfterDashPosition))
    (NCP.CardAddText HistoryCard Time AfterDashPosition)
    (TEDIT.LOOKS (NCP.CardSubstance HistoryCard)
    ' (WEIGHT BOLD)
    AfterDashPosition
    (NCHARS Time])

```

**(NCHist.AddHistoryCard**

[LAMBDA NIL

; Edited 26-Jul-88 10:43 by pmi

;; Add History card type to card type list

;; pmi 10/22/87: Added left button items.

;; pmi 7/26/88: Added Start and Suspend Logging items.

```

(NCP.CreateCardType 'History 'Text `[ (MakeFn , (FUNCTION NCHist.HistoryCardMakeFn]
  `((DisplayedInMenuFlg T)
    (LeftButtonMenuItems , (APPEND (NC.GetCardTypeField LeftButtonMenuItems 'Text)
      ' ("-----" (FUNCTION NULL))
      (Compute% History (FUNCTION NCHist.ComputeHistory)
        "Computes a history from the current log file.")
      (|Change History Event Specs| (FUNCTION
        NCHist.ChangeHistoryEventSpecs)
        "Change event specs used to compute a history.")
      (|Convert Pointers to Links| (FUNCTION NCHist.ConvertPointersToLinks
        )
        "Converts all pointer icons to real links.")
      (Start% Logging (FUNCTION NCHist.StartHistoryCardLogging)
        "Start logging for this History card."
        (SUBITEMS (Suspend% Logging (FUNCTION
          NCHist.SuspendHistoryCardLogging
          )
          "Suspend logging for this History card."))

```

)

(DEFINEQ

**(NCHist.ShaveLeadingSpace**

[LAMBDA (String)

; Edited 20-Jun-88 17:37 by pmi

;;; If first char of String is a space, then return substrng consisting of the rest of String.

```

(if (EQ (NTHCHARCODE String 1)
    32)
  then (SUBSTRING String 2)
  else String])

```

)

;;; General automated history card functions

(DEFINEQ

**(NCHist.CreateHistoryDate**

[LAMBDA (Card)

; Edited 20-Jun-88 17:37 by pmi

;;; Creates date for history card.

```

(LET [(SystemDate (DATE (DATEFORMAT NO.TIME NUMBER.OF.MONTH]
  (CONCAT (NCHist.ShaveLeadingSpace (SUBSTRING SystemDate 4 5))
    "/"
    (NCHist.ShaveLeadingSpace (SUBSTRING SystemDate 1 2))
    "/"
    (NCHist.ShaveLeadingSpace (SUBSTRING SystemDate 7 8])

```

**(NCHist.ComputeHistory**

[LAMBDA (Window)

; Edited 26-Jul-88 10:32 by pmi

;; Computes stuff for History card.

;; pmi 7/26/88: Now checks if logging is turned on for this history card before computing the history.

```

(DECLARE (GLOBALVARS NCHist.HistoryEventHashArray NCHist.HistoryLinkType NC.MsgDelay))
(PROG ((HistoryCard (NCP.CoerceToCard Window))
  NoteFile NCLogInfo ActiveCards ContentsFileBox ToBeFiledFileBox OrphansFileBox)
  (SETQ NoteFile (NCP.CardNoteFile HistoryCard))
  (SETQ NCLogInfo (NCP.CardUserDataProp HistoryCard 'NCLogInfo))
  (if (OR (NULL NCLogInfo)

```

```

    (NOT (NCLOG.LoggingOnP NoteFile NCLogInfo)))
  then (NCP.PrintMsg Window T "Can't compute history - logging is not turned on for this History
        card.")
        (NCP.ClearMsg Window T NC.MsgDelay)
        (RETURN NIL))
  (SETQ ActiveCards (NC.MapCards NoteFile [FUNCTION (LAMBDA (Card)
                                                    Card]
                                           (FUNCTION NC.ActiveCardP)))

;;; Warn the user about changes to unsaved cards not being logged until a checkpoint is done.

(NCP.PrintMsg Window T "Checking for active cards...")
(if [for ActiveCard in ActiveCards thereis (AND (OR (NC.CardDirtyP ActiveCard)
                                                    (NCP.NewCardP ActiveCard))
        (NEQ ActiveCard (fetch (NoteFile LinkLabelsCard)
                                of NoteFile))
        (NEQ ActiveCard (fetch (NoteFile RegistryCard) of NoteFile)
        )
        (NEQ ActiveCard (NCP.LookupCardByName 'OpenEventsCard
                                                NoteFile))
        (NEQ (NCP.CardType ActiveCard)
              'History)
        [NOT (AND (EQ ActiveCard (SETQ ContentsFileBox
                                    (NCP.ContentsFileBox NoteFile)))
                  (NOT (NCP.CardDisplayedP ContentsFileBox]
        [NOT (AND (EQ ActiveCard (SETQ ToBeFiledFileBox
                                    (NCP.ToBeFiledFileBox NoteFile))
                  )
                  (NOT (NCP.CardDisplayedP ToBeFiledFileBox]
        (NOT (AND (EQ ActiveCard (SETQ OrphansFileBox (
                                                        NCP.OrphansFileBox
                                                        NoteFile)))
                  (NOT (NCP.CardDisplayedP OrphansFileBox]
        then (SELECTQ (NC.AskUserWithMenu ' (Yes No Cancel)
          (CONCAT "Warning - information about cards not yet saved will not be
                  logged." (CHARACTER 13)
                  "Checkpoint before computing history?")
          Window NIL T)
          (Yes (NCP.CheckpointNoteFiles NoteFile))
          (No)
          (Cancel (RETURN NIL))
          (RETURN NIL)))
  (RETURN (RESETLST
    [RESETSAVE `(NCLOG.SuspendLogging ,NoteFile ,NCLogInfo)
      `(NCLOG.StartLogging ,NoteFile ,NCLogInfo]
    (LET (LoggingStream SectionEndPtr SectionPtrs HistoryEventTypes)
      (SETQ LoggingStream (fetch (NCLogInfo LoggingStream) of NCLogInfo))
      (SETQ SectionEndPtr (GETEOFPTR LoggingStream))
      (SETQ SectionPtrs (NCP.CardUserDataProp HistoryCard 'SectionPtrs))

```

;;; Compute history for current section if it has not already been computed.

```

  (if (NOT (NCP.ValidLinkTypeP NCHist.HistoryLinkType NoteFile))
    then (NCP.CreateLinkType NCHist.HistoryLinkType NoteFile))
  (if (FMEMB SectionEndPtr SectionPtrs)
    then (NCP.PrintMsg Window T "No new events to report.")
         (NCP.ClearMsg Window T NC.MsgDelay)
    else (NC.PrintMsg Window T "Computing History...")
         (if SectionPtrs
           then (SETQ SectionStartPtr (CAR (LAST SectionPtrs)))
           else (SETQ SectionStartPtr 0))
         (SETQ SectionPtrs (NCONC1 SectionPtrs SectionEndPtr))
         (NCP.CardUserDataProp HistoryCard 'SectionPtrs SectionPtrs)
         (SETQ HistoryEventTypes)

```

;;; Gather up all of the events to be logged for this History card. They are stored on the History card, as separate lists for each event category.

```

[for HistoryEventList in (NCP.CardUserDataProp HistoryCard 'HistoryEventSpecs)
  do (for HistoryEventType in HistoryEventList
    do (SETQ HistoryEventTypes (APPEND (MKLIST (GETHASH HistoryEventType
                                                        NCHist.HistoryEventHashArray
                                                        ))
                                         HistoryEventTypes]
  (SETFILEPTR LoggingStream SectionStartPtr)
  (while (NOT (EOF LoggingStream)) first (SKIPSEPRS LoggingStream)
    finally (NCP.CardAddText HistoryCard (CONCAT (CHARACTER 13)
                                                  " --- "
                                                  (DATE (DATEFORMAT NO.DATE
                                                            NO.SECONDS))
                                                  " ---"
                                                  (CHARACTER 13)
                                                  (CHARACTER 13))
        'END)
    do (NCHist.GetLogEventFromStream LoggingStream HistoryEventTypes HistoryCard
      ))
  (NC.PrintMsg Window NIL "Done.")
  (NCP.ClearMsg Window T NC.MsgDelay))))))

```

**(NCHist.GetLogEventFromStream**

```
[LAMBDA (Stream HistoryEventTypes HistoryCard) ; Edited 20-Jun-88 17:38 by pmi
  (LET (LogEvent EventType)
    (SKIPSEPRS Stream)
    [if (NOT (EOFP Stream))
      then (SETQ LogEvent (READ Stream))
            (SETQ EventType (CAR LogEvent))
            (if (FMEMB EventType HistoryEventTypes)
              then (APPLY (GETHASH EventType NCHist.LogEventFnHashArray)
                          (CONS HistoryCard (CDR LogEvent))
                          LogEvent])])]
```

**(NCHist.ProcessLogEvent**

```
[LAMBDA (HistoryCard EventType EventTypeArgs) ; Edited 20-Jun-88 17:38 by pmi
  (DECLARE (GLOBALVARS NCLOG.UIDHashArray NCHist.LogEventFnHashArray))
  (APPLY (GETHASH EventType NCHist.LogEventFnHashArray)
    (CONS HistoryCard EventTypeArgs))]
```

**(NCHist.ChangeHistoryEventSpecs**

```
[LAMBDA (Window) ; Edited 20-Jun-88 17:38 by pmi
```

```
;;; Change the values of the various History event specs
```

```
(LET ((Card (NCP.CoerceToCard Window)))
  (NC.ProtectedCardOperation Card "History Event Specs" NIL (PROG (HistoryEventSpecs)
                                                                (SETQ HistoryEventSpecs
                                                                  (NCHist.AskHistoryEventSpecs Window
                                                                    Card))
```

```
;;; NC.SetPropListDirtyFlg Card T
```

```
(NCP.CardUserDataProp Card
  'HistoryEventSpecs
  HistoryEventSpecs)
(NC.ClearMsg Window T])
```

**(NCHist.AskHistoryEventSpecs**

```
[LAMBDA (MainWindow HistoryCard) ; Edited 20-Jun-88 17:38 by pmi
```

```
;;; Puts up a stylesheet asking user about events, link types, etc. over which to compute the next history. The stylesheet returns a list of n things: .
```

```
(DECLARE (GLOBALVARS NCHist.HistoryEventsSpecsStylesheet NCHist.EventCategories NC.MenuFont))
(PROG (NoteFile OldEvents Position Choices)
  (SETQ NoteFile (NCP.CardNoteFile HistoryCard))
  (SETQ OldEvents (NCP.CardUserDataProp HistoryCard 'DefaultHistoryEventSpecs))
  (if NCHist.HistoryEventsSpecsStylesheet
    else (NCHist.InitializeHistoryEventsSpecsStylesheet))
  [SETQ Position (AND (WINDOWP MainWindow)
    (create POSITION
      XCOORD _ (fetch (REGION LEFT) of (WINDOWPROP MainWindow 'REGION))
      YCOORD _ (fetch (REGION TOP) of (WINDOWREGION MainWindow))
      ; The stylesheet is in a global var. We only need to provide its
      ; position, items, and selections.
      (STYLE.PROP NCHist.HistoryEventsSpecsStylesheet 'POSITION Position)
      (STYLE.PROP NCHist.HistoryEventsSpecsStylesheet 'ITEMS
        (for EventCategory in NCHist.EventCategories
          collect (create MENU
            ITEMS _ (if (fetch (EventCategory MenuItems) of EventCategory)
                      else (APPLY* (fetch (EventCategory MenuItemsFn) of EventCategory)
                                   NoteFile))
            MENUFONT _ NC.MenuFont)))
      [STYLE.PROP NCHist.HistoryEventsSpecsStylesheet 'SELECTIONS
        (OR OldEvents (for EventCategory in NCHist.EventCategories bind DefaultMenuItemsFn
          collect (if (fetch (EventCategory DefaultMenuItems) of EventCategory)
                    elseif (SETQ DefaultMenuItemsFn (fetch (EventCategory DefaultMenuItemsFn
                                                              of EventCategory))
                      then (APPLY* DefaultMenuItemsFn NoteFile]
        (if (SETQ Choices (STYLESHEET NCHist.HistoryEventsSpecsStylesheet))
          then (NCP.CardUserDataProp HistoryCard 'DefaultHistoryEventSpecs Choices)
              (RETURN (for CategoryChoices in Choices as EventCategory in NCHist.EventCategories bind
                MenuChoicesFn
                collect (if (SETQ MenuChoicesFn (fetch (EventCategory MenuChoicesFn) of EventCategory))
                          then (APPLY* MenuChoicesFn CategoryChoices HistoryCard NoteFile)
                          else CategoryChoices)))
          else (RETURN OldEvents)])]
```

**(NCHist.InitializeHistoryEventsSpecsStylesheet**

```
[LAMBDA NIL ; Edited 20-Jun-88 17:39 by pmi
```

```
;;; pmi 10/30/87: Build a stylesheet for the History Event Specs, using the events stored by category in the ASSOC list NCLOG.ListOfEventTypes.
```

```
(DECLARE (GLOBALVARS NCHist.HistoryEventsSpecsStylesheet NCHist.EventCategories NC.MenuFont))
(LET (NumCategories
  (SETQ NumCategories (LENGTH NCHist.EventCategories))
  (SETQ NCHist.HistoryEventsSpecsStylesheet (CREATE.STYLE 'ITEMS (from 1 to NumCategories
    collect (create MENU
      ITEMS _ T))
    'SELECTIONS
    (from 1 to NumCategories collect T)
    'ITEM.TITLES
    (for EventCategory in NCHist.EventCategories
      collect (fetch (EventCategory Name) of EventCategory))
    'ITEM.TITLE.FONT
    (FONTCOPY NC.MenuFont 'WEIGHT 'BOLD)
    'NEED.NOT.FILL.IN
    (from 1 to NumCategories collect 'MULTI)
    'TITLE "History Event Specs?"]]))
```

**(NCHist.InitializeHistoryEventHashArray**

```
[LAMBDA NIL
  (DECLARE (GLOBALVARS NCHist.HistoryEventHashArray NCLOG.HashArraySize))
  ; Edited 20-Jun-88 17:39 by pmi
  (if (HARRAYP NCHist.HistoryEventHashArray)
    else (SETQ NCHist.HistoryEventHashArray (HASHARRAY NCLOG.HashArraySize)))
  (PUTHASH 'Create% Card 'GetNewCard NCHist.HistoryEventHashArray)
  (PUTHASH '|Bring Up Card| 'EditCard.End NCHist.HistoryEventHashArray)
  (PUTHASH 'Close% Card 'QuitCard.End NCHist.HistoryEventHashArray)
  (PUTHASH '|Change Card Substance| 'PutMainCardData NCHist.HistoryEventHashArray)
  (PUTHASH '|Change Card Prop List| 'PutPropList NCHist.HistoryEventHashArray)
  (PUTHASH '|Change Card Title| ' (AssignTitle.Begin PutTitle)
    NCHist.HistoryEventHashArray)
  (PUTHASH 'Delete% Card 'DelCard NCHist.HistoryEventHashArray)
  (PUTHASH 'Create% Link 'MakeLink NCHist.HistoryEventHashArray)
  (PUTHASH 'Delete% Link 'DelLink NCHist.HistoryEventHashArray)
  (PUTHASH 'Recompute% Browser 'RecomputeBrowser.End NCHist.HistoryEventHashArray)
  (PUTHASH 'Relayout% Browser 'RelayoutBrowser.End NCHist.HistoryEventHashArray)
  (PUTHASH '|Add Card Type| 'AddCardType NCHist.HistoryEventHashArray)
  NCHist.HistoryEventHashArray])
```

**(NCHist.InitializeLogEventFnHashArray**

```
[LAMBDA NIL
  ; Edited 20-Jun-88 17:39 by pmi
  (DECLARE (GLOBALVARS NCHist.LogEventFnHashArray NCLOG.HashArraySize))
  (if (HARRAYP NCHist.LogEventFnHashArray)
    else (SETQ NCHist.LogEventFnHashArray (HASHARRAY NCLOG.HashArraySize)))
  (PUTHASH 'GetNewCard 'NCHist.GetNewCardDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'EditCard.End 'NCHist.EditCard.EndDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'QuitCard.End 'NCHist.QuitCard.EndDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'PutMainCardData 'NCHist.PutMainCardDataDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'PutPropList 'NCHist.PutPropListDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'AssignTitle.Begin 'NCHist.AssignTitle.BeginDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'PutTitle 'NCHist.PutTitleDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'DelCard 'NCHist.DelCardDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'MakeLink 'NCHist.MakeLinkDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'DelLink 'NCHist.DelLinkDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'RecomputeBrowser.End 'NCHist.RecomputeBrowser.EndDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'RelayoutBrowser.End 'NCHist.RelayoutBrowser.EndDisplayFn NCHist.LogEventFnHashArray)
  (PUTHASH 'AddCardType 'NCHist.AddCardTypeDisplayFn NCHist.LogEventFnHashArray)
  NCHist.LogEventFnHashArray])
```

**(NCHist.InitializeEventCategories**

```
[LAMBDA NIL
  (DECLARE (GLOBALVARS NCHist.EventCategories))
  ; Edited 20-Jun-88 16:36 by pmi
  (SETQ NCHist.EventCategories (LIST NCHist.CardEvents NCHist.CreateLinkEvents NCHist.DeleteLinkEvents
    NCHist.BrowserEvents NCHist.CardTypeEvents]))
```

**(NCHist.ConvertPointersToLinks**

```
[LAMBDA (Window)
  ; Edited 20-Jun-88 17:39 by pmi
  (LET (PointerIcons
    (SETQ PointerIcons (TEDIT.LIST.OF.OBJECTS (TEXTOBJ Window)
      (FUNCTION NC.PointerIconImageObjP)))
    (for PointerIcon in PointerIcons do (NC.ConvertPointerIconToLinkIcon Window (CAR PointerIcon)))
    Window)])
```

**(NCHist.StartLogging**

```
[LAMBDA (Card NLogInfo)
  ; Edited 13-Jul-88 17:38 by pmi
  ;; Start up logging for this History card.
  ;; pmi 7/12/88: Revamped to do more work on setting up a History card for logging.
  (DECLARE (GLOBALVARS NCHist.EventCategories))
  (LET ((NoteFile (NCP.CardNoteFile Card))
    (if (SETQ NLogInfo (NCLOG.StartLogging NoteFile NLogInfo))
      then (NCP.CardUserDataProp Card 'NLogInfo NLogInfo)))
```

```

(NCP.CardUserDataProp Card 'HistoryEventSpecs
  (for EventCategory in NCHist.EventCategories bind CategoryChoices DefaultMenuItemsFn
    MenuChoicesFn
      collect (SETQ CategoryChoices (if (fetch (EventCategory DefaultMenuItems) of
                                                )
                                          )
              elseif (SETQ DefaultMenuItemsFn (fetch (EventCategory
                                                        DefaultMenuItemsFn
                                                          )
                                                       of EventCategory))
                then (APPLY* DefaultMenuItemsFn NoteFile)))
  (if (SETQ MenuChoicesFn (fetch (EventCategory MenuChoicesFn) of EventCategory))
      then (APPLY* MenuChoicesFn CategoryChoices Card NoteFile)
      else CategoryChoices)))
NCLogInfo])

```

**(NCHist.SuspendLogging**

[LAMBDA (CardOrNoteFile NCLogInfo)

; Edited 13-Jul-88 15:17 by pmi

```

;; Suspend logging for this notefile.
(if (type? Card CardOrNoteFile)
    then (NCLog.SuspendLogging (NCP.CardNoteFile CardOrNoteFile)
                                NCLogInfo)
    elseif (type? NoteFile CardOrNoteFile)
        then (NCLog.SuspendLogging CardOrNoteFile NCLogInfo])

```

**(NCHist.StopLogging**

[LAMBDA (CardOrNoteFile NCLogInfo)

; Edited 13-Jul-88 15:27 by pmi

```

;; Stop logging for this notefile.
;; pmi 7/12/88: now takes CardOrNoteFile argument, instead of card.
(if (type? Card CardOrNoteFile)
    then (NCLog.StopLogging (NCP.CardNoteFile CardOrNoteFile)
                             NCLogInfo)
    elseif (type? NoteFile CardOrNoteFile)
        then (NCLog.StopLogging CardOrNoteFile NCLogInfo])

```

**(NCHist.StartNoteFileLogging**

[LAMBDA (NoteFile)

; Edited 26-Jul-88 10:48 by pmi

```

;; Start up logging on all active, logging History cards in this notefile.
(DECLARE (GLOBALVARS NC.MsgDelay))
(LET ((Window (NCP.CoerceToInterestedWindow NoteFile))
      NCLogInfo)
  (NCP.PrintMsg Window T "Starting logging...")
  (for HistoryCard in (NCP.NoteFileProp NoteFile 'HistoryCards)
    when (AND (NCP.ValidCardP HistoryCard)
              (SETQ NCLogInfo (NCP.CardUserDataProp HistoryCard 'NCLogInfo))
              (NOT (NCLog.LoggingOnP NoteFile NCLogInfo)))
    do (NCHist.StartLogging HistoryCard NCLogInfo))
  (NCP.PrintMsg Window NIL "done.")
  (NCP.ClearMsg Window T NC.MsgDelay])

```

**(NCHist.SuspendNoteFileLogging**

[LAMBDA (NoteFile)

; Edited 26-Jul-88 10:49 by pmi

```

;; Suspend logging on all active, logging History cards in this notefile.
(DECLARE (GLOBALVARS NC.MsgDelay))
(LET ((Window (NCP.CoerceToInterestedWindow NoteFile))
      NCLogInfo)
  (NCP.PrintMsg Window T "Suspending logging...")
  (for HistoryCard in (NCP.NoteFileProp NoteFile 'HistoryCards) when (AND (NCP.ValidCardP HistoryCard)
                                                                              (SETQ NCLogInfo
                                                                              (NCP.CardUserDataProp
                                                                              HistoryCard
                                                                              'NCLogInfo))
                                                                              (NCLog.LoggingOnP NoteFile
                                                                              NCLogInfo))
    do (NCHist.SuspendLogging HistoryCard NCLogInfo))
  (NCP.PrintMsg Window NIL "done.")
  (NCP.ClearMsg Window T NC.MsgDelay])

```

**(NCHist.StartHistoryCardLogging**

[LAMBDA (Window)

; Edited 13-Jul-88 17:34 by pmi

```

;; If this history card is not currently logging, turn logging on.
(DECLARE (GLOBALVARS NC.MsgDelay))
(LET ((HistoryCard (NCP.CardFromWindow Window))
      NCLogInfo)
  (SETQ NCLogInfo (NCP.CardUserDataProp HistoryCard 'NCLogInfo))
  (if (NOT (NCLog.LoggingOnP (NCP.CardNoteFile HistoryCard)
                              NCLogInfo))
      (NCHist.StartLogging HistoryCard NCLogInfo)
      (NCHist.StopLogging HistoryCard NCLogInfo))

```



```

        NCLogInfo))
    then (NCHist.StartLogging HistoryCard NCLogInfo)
        (NCP.PrintMsg Window T "Logging started.")
        (NCP.ClearMsg Window T NC.MsgDelay)
    else (NCP.PrintMsg Window T "Logging already on.")
        (NCP.ClearMsg Window T NC.MsgDelay])

```

**(NCHist.SuspendHistoryCardLogging**

[LAMBDA (Window)

; Edited 13-Jul-88 17:33 by pmi

;; If this history card is currently logging, suspend logging.

```

    (DECLARE (GLOBALVARS NC.MsgDelay))
    (LET ((HistoryCard (NCP.CardFromWindow Window))
        NCLogInfo)
        (if (AND (SETQ NCLogInfo (NCP.CardUserDataProp HistoryCard 'NCLogInfo))
            (NCLOG.LoggingOnP (NCP.CardNoteFile HistoryCard)
                NCLogInfo))
            then (NCHist.SuspendLogging HistoryCard (NCP.CardUserDataProp HistoryCard 'NCLogInfo))
                (NCP.PrintMsg Window T "Logging suspended.")
                (NCP.ClearMsg Window T NC.MsgDelay)
            else (NCP.PrintMsg Window T "Logging already suspended.")
                (NCP.ClearMsg Window T NC.MsgDelay]))

```

)

;;; Fns for the various event category functions

(DEFINEQ

**(NCHist.CreateLinkEventsMenuItemsFn**

```

    [LAMBDA (NoteFile)
      (REVERSE (NCP.LinkTypes NoteFile))]

```

; Edited 20-Jun-88 17:40 by pmi

**(NCHist.CreateLinkEventsDefaultMenuChoicesFn**

```

    [LAMBDA (NoteFile)
      (REMOVE 'History (APPEND (NCP.UserLinkTypes NoteFile)
        ' (FiledCard SubBox))]

```

; Edited 20-Jun-88 17:40 by pmi

**(NCHist.CreateLinkEventsMenuChoicesFn**

```

    [LAMBDA (Choices HistoryCard NoteFile)
      (NCP.CardUserDataProp HistoryCard 'CreateLinkTypesToLog Choices)
      ' (Create% Link)]

```

; Edited 20-Jun-88 17:40 by pmi

**(NCHist.DeleteLinkEventsMenuItemsFn**

```

    [LAMBDA (NoteFile)
      (REVERSE (NCP.LinkTypes NoteFile))]

```

; Edited 20-Jun-88 17:41 by pmi

**(NCHist.DeleteLinkEventsDefaultMenuChoicesFn**

```

    [LAMBDA (NoteFile)
      (REMOVE 'History (APPEND (NCP.UserLinkTypes NoteFile)
        ' (FiledCard SubBox))]

```

; Edited 20-Jun-88 17:41 by pmi

**(NCHist.DeleteLinkEventsMenuChoicesFn**

```

    [LAMBDA (Choices HistoryCard NoteFile)
      (NCP.CardUserDataProp HistoryCard 'DeleteLinkTypesToLog Choices)
      ' (Delete% Link)]

```

; Edited 20-Jun-88 17:41 by pmi

)

;;; Fns called to display the events in the History Card

(DEFINEQ

**(NCHist.EditCard.BeginDisplayFn**

[LAMBDA (HistoryCard EventTime NoteFile Card)]

**(NCHist.EditCard.EndDisplayFn**

```

    [LAMBDA (HistoryCard EventTime NoteFileNum CardNum)
      (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
      (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
        (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
          NoteFile)))
        (if (NEQ Card HistoryCard)
            then (NCP.CardAddText HistoryCard "Brought up " 'END)
                [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
                    (NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
                    (NCP.CardAddText HistoryCard (CHARACTER 13)
                        'END])

```

; Edited 13-Apr-88 15:33 by pmi

**(NCHist.QuitCard.BeginDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card Don'tDeactivateFlg)]
```

**(NCHist.QuitCard.EndDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardNum Don'tDeactivateFlg)
; Edited 13-Apr-88 15:34 by pmi
(DECLARE (GLOBALVARS NCHist.HistoryLinkType))
(LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
(Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
NoteFile)))
(if (NEQ Card HistoryCard)
then (NCP.CardAddText HistoryCard "Closed " 'END)
[NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
(NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
(NCP.CardAddText HistoryCard (CHARACTER 13)
'END]]])
```

**(NCHist.MakeCard.BeginDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardType Title NewCardNum)
; Edited 20-Jun-88 17:42 by pmi
NIL])
```

**(NCHist.MakeCard.EndDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardType Title CardNum)
; Edited 13-Apr-88 15:34 by pmi
(DECLARE (GLOBALVARS NCHist.HistoryLinkType))
(LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
(Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
NoteFile)))
(if (NEQ Card HistoryCard)
then (NCP.CardAddText HistoryCard "Created " 'END)
(if CardType
then (NCP.CardAddText HistoryCard (MKSTRING CardType)
'END))
(NCP.CardAddText HistoryCard " card " 'END)
[NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
(NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
(if Title
then (NCP.CardAddText HistoryCard (CONCAT " with title " Title)
'END))
(NCP.CardAddText HistoryCard (CHARACTER 13)
'END]]])
```

**(NCHist.GetNewCardDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardNum CardType) ; Edited 13-Apr-88 15:34 by pmi
(DECLARE (GLOBALVARS NCHist.HistoryLinkType))
(LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
(Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
NoteFile)))
(if (AND Card (NEQ Card HistoryCard))
then (NCP.CardAddText HistoryCard "Created " 'END)
(if CardType
then (NCP.CardAddText HistoryCard (MKSTRING CardType)
'END))
(NCP.CardAddText HistoryCard " card " 'END)
[NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
(NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
(NCP.CardAddText HistoryCard (CHARACTER 13)
'END]]])
```

**(NCHist.DelCardDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card Title) ; Edited 20-Jun-88 17:42 by pmi
(if (NEQ Card HistoryCard)
then (NCP.CardAddText HistoryCard (CONCAT "Deleted card " Title " " (CHARACTER 13))
'END))
```

**(NCHist.CacheCardDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card) ; Edited 20-Jun-88 17:42 by pmi
(NCP.CardAddText HistoryCard (CONCAT "Cached card " Card (CHARACTER 13]))
```

**(NCHist.UncacheCardDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card) ; Edited 20-Jun-88 17:43 by pmi
(NCP.CardAddText HistoryCard (CONCAT "Uncached card " Card (CHARACTER 13]))
```

**(NCHist.SaveCard.BeginDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card)]
```

**(NCHist.SaveCard.EndDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card) ; Edited 20-Jun-88 17:43 by pmi
  (NCP.CardAddText HistoryCard (CONCAT "Changed card " Card (CHARACTER 13]))
```

**(NCHist.PutLinksDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card) ; Edited 20-Jun-88 17:43 by pmi
  (NCP.CardAddText HistoryCard (CONCAT "Put links for card " Card (CHARACTER 13]))
```

**(NCHist.PutMainCardDataDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardNum) ; Edited 13-Apr-88 15:35 by pmi
  (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
  (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
    (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
      NoteFile)))
    (if (AND Card (NEQ Card HistoryCard)
      (NEQ Card (fetch (NoteFile LinkLabelsCard) of NoteFile))
      (NEQ Card (fetch (NoteFile RegistryCard) of NoteFile))
      (NEQ Card (NCP.LookupCardByName 'OpenEventsCard NoteFile)))
      then (NCP.CardAddText HistoryCard "Changed contents of " 'END)
        [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
          (NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
          (NCP.CardAddText HistoryCard (CHARACTER 13)
            'END]]
```

**(NCHist.PutPropListDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardNum) ; Edited 13-Apr-88 15:35 by pmi
  (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
  (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
    (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
      NoteFile)))
    (if (AND Card (NEQ Card HistoryCard)
      (NEQ Card (fetch (NoteFile LinkLabelsCard) of NoteFile))
      (NEQ Card (fetch (NoteFile RegistryCard) of NoteFile))
      (NEQ Card (NCP.LookupCardByName 'OpenEventsCard NoteFile)))
      then (NCP.CardAddText HistoryCard "Changed property list of " 'END)
        [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
          (NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
          (NCP.CardAddText HistoryCard (CHARACTER 13)
            'END]]
```

**(NCHist.PutTitleDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum CardNum) ; Edited 20-Jun-88 12:28 by pmi
  (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
  (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
    (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
      NoteFile))
    OldTitle)
    (if (AND Card (NEQ Card HistoryCard)
      (NEQ Card (fetch (NoteFile LinkLabelsCard) of NoteFile))
      (NEQ Card (fetch (NoteFile RegistryCard) of NoteFile))
      (NEQ Card (NCP.LookupCardByName 'OpenEventsCard NoteFile)))
      then (NCP.CardAddText HistoryCard "Changed title of " 'END)
        [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
          (NCP.CoerceToLinkDisplayMode ' (FLOAT FLOAT FLOAT)
          (if (SETQ OldTitle (NCP.CardUserDataProp HistoryCard (MKATOM CardNum)))
            then (NCP.CardAddText HistoryCard (CONCAT " from " OldTitle " " to " (NCP.CardTitle Card)
              ""))
            'END)
          (NCP.CardUserDataProp HistoryCard (MKATOM CardNum)
            NIL))
        (NCP.CardAddText HistoryCard (CHARACTER 13)
          'END]]
```

**(NCHist.AssignTitle.BeginDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile CardNum OldTitle) ; Edited 20-Jun-88 17:45 by pmi
  (if (NCP.CardUserDataProp HistoryCard (MKATOM CardNum))
    else (NCP.CardUserDataProp HistoryCard (MKATOM CardNum)
      OldTitle])
```

**(NCHist.AssignTitle.EndDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFile Card NewTitle) ; Edited 20-Jun-88 17:45 by pmi
  (NCP.CardAddText HistoryCard (CONCAT " to " NewTitle (CHARACTER 13]))
```

**(NCHist.MakeLinkDisplayFn**

```
[LAMBDA (HistoryCard EventTime NoteFileNum LinkNum SourceCardNum DestinationCardNum Label AnchorMode
  DisplayMode) ; Edited 13-Apr-88 15:36 by pmi
  (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
  (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
    (SourceCard (NC.CardFromUID (NCLOG.UIDFromNum SourceCardNum)
```

```

        NoteFile))
    (DestinationCard (NC.CardFromUID (NCLOG.UIDFromNum DestinationCardNum)
        NoteFile)))
    (if [AND (NEQ SourceCard HistoryCard)
        (MEMBER Label (NCP.CardUserDataProp HistoryCard 'CreateLinkTypesToLog]
    then (NCP.CardAddText HistoryCard (CONCAT "Created " Label " link from ")
        'END)
        [NC.InsertPointerInText HistoryCard SourceCard NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard " to " 'END)
        [NC.InsertPointerInText HistoryCard DestinationCard NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard (CHARACTER 13)
        'END])

```

**(NCHist.DelLinkDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFileNum LinkNum SourceCardNum DestinationCardNum Label AnchorMode
    DisplayMode)
    ; Edited 13-Apr-88 15:36 by pmi
    (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
    (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
        (SourceCard (NC.CardFromUID (NCLOG.UIDFromNum SourceCardNum)
            NoteFile))
        (DestinationCard (NC.CardFromUID (NCLOG.UIDFromNum DestinationCardNum)
            NoteFile)))
    (if [AND (NEQ SourceCard HistoryCard)
        (MEMBER Label (NCP.CardUserDataProp HistoryCard 'DeleteLinkTypesToLog]
    then (NCP.CardAddText HistoryCard (CONCAT "Deleted " Label " link from ")
        'END)
        [NC.InsertPointerInText HistoryCard SourceCard NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard " to " 'END)
        [NC.InsertPointerInText HistoryCard DestinationCard NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard (CHARACTER 13)
        'END])

```

**(NCHist.RecomputeBrowser.BeginDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFile Card)]

```

**(NCHist.RecomputeBrowser.EndDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFileNum CardNum)
    ; Edited 13-Apr-88 15:36 by pmi
    (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
    (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
        (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
            NoteFile)))
    (if (NEQ Card HistoryCard)
    then (NCP.CardAddText HistoryCard "Recomputed browser " 'END)
        [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard (CHARACTER 13)
        'END])

```

**(NCHist.RelayoutBrowser.BeginDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFile Card)]

```

**(NCHist.RelayoutBrowser.EndDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFileNum CardNum)
    ; Edited 13-Apr-88 15:37 by pmi
    (DECLARE (GLOBALVARS NCHist.HistoryLinkType))
    (LET* ((NoteFile (NC.NoteFileFromNoteFileUID (NCLOG.UIDFromNum NoteFileNum)))
        (Card (NC.CardFromUID (NCLOG.UIDFromNum CardNum)
            NoteFile)))
    (if (NEQ Card HistoryCard)
    then (NCP.CardAddText HistoryCard "Re-laid out " 'END)
        [NC.InsertPointerInText HistoryCard Card NCHist.HistoryLinkType
        (NCP.CoerceToLinkDisplayMode '(FLOAT FLOAT FLOAT)
        (NCP.CardAddText HistoryCard (CHARACTER 13)
        'END])

```

**(NCHist.AddCardTypeDisplayFn**

```

[LAMBDA (HistoryCard EventTime TypeName)
    ; Edited 20-Jun-88 17:46 by pmi
    (NCP.CardAddText HistoryCard (CONCAT "Added card type " TypeName (CHARACTER 13))
    'END])

```

**(NCHist.StartLoggingDisplayFn**

```

[LAMBDA (HistoryCard EventTime NoteFile FullNoteFileName)]

```

```

)

```

```

(DECLARE%: EVAL@COMPILE

```

```
(DATATYPE EventCategory (Name MenuItems MenuItemsFn DefaultMenuItems DefaultMenuItemsFn MenuChoicesFn))
)
```

```
(/DECLAREDATATYPE 'EventCategory' (POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
'12)
```

```
;;; Global variables settable by the user
```

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

```
(GLOBALVARS NCHist.HistoryCardPosition NCHist.InhibitInitialLogging)
)
```

```
(RPAQ? NCHist.HistoryCardPosition NIL)
```

```
(RPAQ? NCHist.InhibitInitialLogging NIL)
```

```
;;; Other global variables
```

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

```
(GLOBALVARS NCHist.HistoryEventsSpecsStylesheet NCHist.HistoryEventHashArray NCHist.LogEventFnHashArray
NCHist.HistoryLinkType NCHist.EventCategories NCHist.CardEvents NCHist.CreateLinkEvents
NCHist.DeleteLinkEvents NCHist.BrowserEvents NCHist.CardTypeEvents)
)
```

```
(RPAQQ NCHist.HistoryLinkType History)
```

```
(RPAQ NCHist.HistoryEventHashArray (HASHARRAY NCLOG.HashArraySize))
```

```
(RPAQ NCHist.LogEventFnHashArray (HASHARRAY NCLOG.HashArraySize))
```

```
(RPAQ NCHist.CardEvents
```

```
(create EventCategory Name _ 'Card% Events MenuItems _ '(Create% Card |Bring Up Card| Close% Card
|Change Card Substance|
|Change Card Prop List| |Change Card Title|
Delete% Card)
MenuItemsFn _ NIL DefaultMenuItems _ '(Create% Card |Change Card Substance| |Change Card Title|
Delete% Card)
DefaultMenuItemsFn _ NIL MenuChoicesFn _ NIL))
```

```
(RPAQ NCHist.CreateLinkEvents (create EventCategory Name _ '|Create Link Events| MenuItems _ NIL MenuItemsFn _
(FUNCTION NCHist.CreateLinkEventsMenuItemsFn)
DefaultMenuItems _ NIL DefaultMenuItemsFn _ (FUNCTION
NCHist.CreateLinkEventsDefaultMenuChoicesFn
)
MenuChoicesFn _ (FUNCTION NCHist.CreateLinkEventsMenuChoicesFn)))
```

```
(RPAQ NCHist.DeleteLinkEvents (create EventCategory Name _ '|Delete Link Events| MenuItems _ NIL MenuItemsFn _
(FUNCTION NCHist.DeleteLinkEventsMenuItemsFn)
DefaultMenuItems _ NIL DefaultMenuItemsFn _ (FUNCTION
NCHist.DeleteLinkEventsDefaultMenuChoicesFn
)
MenuChoicesFn _ (FUNCTION NCHist.DeleteLinkEventsMenuChoicesFn)))
```

```
(RPAQ NCHist.BrowserEvents (create EventCategory Name _ 'Browser% Events MenuItems _ '(Recompute% Browser
Relayout% Browser)
MenuItemsFn _ NIL DefaultMenuItems _ '(Recompute% Browser)
DefaultMenuItemsFn _ NIL MenuChoicesFn _ NIL))
```

```
(RPAQ NCHist.CardTypeEvents (create EventCategory Name _ '|Card Type Events| MenuItems _ '(|Add Card Type|)
MenuItemsFn _ NIL DefaultMenuItems _ NIL DefaultMenuItemsFn _ NIL MenuChoicesFn
_ NIL))
```

```
(NCHist.AddHistoryCard)
```

```
(PUTPROP 'History 'WhenSavedFn (FUNCTION NCHist.HistoryCardWhenSavedFn))
```

```
(NCHist.InitializeHistoryEventsSpecsStylesheet)
```

```
(NCHist.InitializeHistoryEventHashArray)
```

```
(NCHist.InitializeLogEventFnHashArray)
```

```
(NCHist.InitializeEventCategories)
```

```
(PUTPROPS NCHISTORYCARD COPYRIGHT ("Xerox Corporation" 1986 1987 1988))
```

---

## FUNCTION INDEX

NCHist.AddCardTypeDisplayFn .....	12	NCHist.InitializeHistoryEventsSpecsStylesheet .....	6
NCHist.AddHistoryCard .....	4	NCHist.InitializeLogEventFnHashArray .....	7
NCHist.AskHistoryEventSpecs .....	6	NCHist.MakeCard.BeginDisplayFn .....	10
NCHist.AssignTitle.BeginDisplayFn .....	11	NCHist.MakeCard.EndDisplayFn .....	10
NCHist.AssignTitle.EndDisplayFn .....	11	NCHist.MakeLinkDisplayFn .....	11
NCHist.CacheCardDisplayFn .....	10	NCHist.NoteFileWhenOpenedFn .....	2
NCHist.ChangeHistoryEventSpecs .....	6	NCHist.ProcessLogEvent .....	6
NCHist.ComputeHistory .....	4	NCHist.PutLinksDisplayFn .....	11
NCHist.ConvertPointersToLinks .....	7	NCHist.PutMainCardDataDisplayFn .....	11
NCHist.CreateHistoryCardTitle .....	3	NCHist.PutPropListDisplayFn .....	11
NCHist.CreateHistoryDate .....	4	NCHist.PutTitleDisplayFn .....	11
NCHist.CreateLinkEventsDefaultMenuChoicesFn .....	9	NCHist.QuitCard.BeginDisplayFn .....	10
NCHist.CreateLinkEventsMenuChoicesFn .....	9	NCHist.QuitCard.EndDisplayFn .....	10
NCHist.CreateLinkEventsMenuItemsFn .....	9	NCHist.RecomputeBrowser.BeginDisplayFn .....	12
NCHist.DelCardDisplayFn .....	10	NCHist.RecomputeBrowser.EndDisplayFn .....	12
NCHist.DeleteLinkEventsDefaultMenuChoicesFn .....	9	NCHist.RelayoutBrowser.BeginDisplayFn .....	12
NCHist.DeleteLinkEventsMenuChoicesFn .....	9	NCHist.RelayoutBrowser.EndDisplayFn .....	12
NCHist.DeleteLinkEventsMenuItemsFn .....	9	NCHist.SaveCard.BeginDisplayFn .....	10
NCHist.DelLinkDisplayFn .....	12	NCHist.SaveCard.EndDisplayFn .....	11
NCHist.EditCard.BeginDisplayFn .....	9	NCHist.ShaveLeadingSpace .....	4
NCHist.EditCard.EndDisplayFn .....	9	NCHist.StartHistoryCardLogging .....	8
NCHist.GetHistoryFileBox .....	3	NCHist.StartLogging .....	7
NCHist.GetLogEventFromStream .....	6	NCHist.StartLoggingDisplayFn .....	12
NCHist.GetNewCardDisplayFn .....	10	NCHist.StartNoteFileLogging .....	8
NCHist.HistoricizeNoteFile .....	2	NCHist.StopLogging .....	8
NCHist.HistoryCardMakeFn .....	3	NCHist.SuspendHistoryCardLogging .....	9
NCHist.HistoryCardWhenSavedFn .....	3	NCHist.SuspendLogging .....	8
NCHist.InitializeEventCategories .....	7	NCHist.SuspendNoteFileLogging .....	8
NCHist.InitializeHistoryEventHashArray .....	7	NCHist.UncacheCardDisplayFn .....	10

---

## VARIABLE INDEX

NCHist.BrowserEvents .....	13	NCHist.DeleteLinkEvents .....	13	NCHist.InhibitInitialLogging ....	13
NCHist.CardEvents .....	13	NCHist.HistoryCardPosition .....	13	NCHist.LogEventFnHashArray .....	13
NCHist.CardTypeEvents .....	13	NCHist.HistoryEventHashArray ....	13	NCHISTORYCARDCOMS .....	1
NCHist.CreateLinkEvents .....	13	NCHist.HistoryLinkType .....	13		

---

## RECORD INDEX

EventCategory .....	13
---------------------	----

---

## PROPERTY INDEX

NCHISTORYCARD .....	2
---------------------	---

---