

File created: 7-Mar-2024 22:40:22 {DSK}<home>frank<il>notecards>system>NCINTERFACE.;2

changes to: (FNS NC.MakeTEditReadOnly NC.MakeTEditReadWrite)

previous date: 8-Oct-2021 13:49:41 {DSK}<home>frank<il>notecards>system>NCINTERFACE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ NCINTERFACECOMS  
(

;;; Internal variables

```
(GLOBALVARS NC.NoticedNoteFilesMenuNewItem NC.NoticedNoteFilesMenu NC.CachedMenus
NC.NoteCardsIconPosition NC.SelectionInProgress NC.SelectedCards NC.SelectingSourcesMenu
NC.SelectingContentsMenu NC.SelectingSingleCardMenu NC.SelectingMultipleCardsMenu
NC.DeleteSelectingMenu NC.SelectingParentsMenu NC.SelectingBrowserSourceMenu
NC.SelectingFileBoxChildrenMenu NC.SelectingCardsMenu NC.SelectingCardMenu NC.TEditMenus
NC.NoteCardTypeMenu NC.DatabaseOperationsMenu NC.LinkLabelMenu NC.WindowRightButtonMenu
NC.LogoWindow LOGOW ExecWindow PROCESS.STATUS.WINDOW QUADTREE.HIGHRES NC.BrowseOrSearchMenu
FONTDIRECTORIES NC.DatabaseFileNameSuggestion NC.UncachingNotCompleted NC.UnspecifiedLinkLabel
NC.UCASESystemLinkLabels NC.MainMenuPosition NC.NoteCardsIconWindow NC.NewCardsIconBitmap
NC.BoxesIconBitmap NC.NoteFileIndicatorFont NC.NoteFileMenuLingerFlg
NC.FileBrowserExpungeMenuItem NC.FileBrowserRecomputeMenuItem NC.FileBrowserMenuItemsToKeep
NC.DefaultNoteFileIconMiddleButtonItem NC.OpenedNoteFileMenu NC.ClosedNoteFileMenu
NC.NoteFileIconOperationsMenuItems NC.NoteFileIconOpenOperations NC.NoteFileIconCloseOperations
NC.NoteCardTypeMenu NC.DefaultLeftButtonMenuItems NC.LocalInsertLinkMenuItem
NC.GlobalInsertLinkMenuItem NC.DefaultMiddleButtonMenuItems NC.PromptWindowMonitorLock)
(INITVARS (NC.LinkLabelMenu NIL)
(NC.LinkDisplayModesMenu NIL)
(NC.DatabaseFileNameSuggestion NIL)
(NC.TEditMenus NIL)
(NC.OpenedNoteFileMenu NIL)
(NC.ClosedNoteFileMenu NIL)
(NC.NoteCardsIconPosition NIL)
(NC.PromptWindowMonitorLock (CREATE.MONITORLOCK "PromptWindowMonitorLock"))
(NC.FileNameFont (FONTCREATE 'HELVETICA 10 'BOLD))
(NC.FullFileNameFont (FONTCREATE 'HELVETICA 8 'STANDARD))
(NC.MenuTitleFont (FONTCREATE 'HELVETICA 10 'BOLD))
(NC.MaxFileNameChars 18))
```

;;; Setup and top level interaction

```
[DECLARE%: DONTVAL@LOAD DONTCOPY EVAL@COMPILE (P (LOAD? (NC.FindFile 'FILEBROWSER NIL NIL T
' LISPUSERSDIRECTORIES)
'PROP])
(FNS NoteCards NC.MakeMainMenu NC.LogoutAdvice NC.AttachPromptWindow NC.AttachPromptWindowOffScreen
NC.GETPROMPTWINDOW NC.MaxPromptWindowWidth NC.PromptWindowOpenP)
```

;;; Menus of all sorts

```
(INITVARS (NC.CachedMenus ' (NC.NoteCardTypeMenu NC.LinkIconMiddleButtonMenu NC.CardOpsMenu
NC.NoteFileOpsMenu NC.OtherOpsMenu))
(NC.DesiredTypeOrdering ' (Text FileBox Sketch Graph BAR Browser Search LinkIndex Document))
(NC.NoticedNoteFilesMenu)
(NC.NoticedNoteFilesMenuNewItem))
(FNS NC.AskLinkLabel NC.AskNoteCardType NC.SortCardTypes NC.ExpandBars NC.ChooseTopLevelCard
NC.RemoveSketchMenuItems NC.SelectionMenusWhenSelectedFn NC.InstallTitleBarButtonEventFn
NC.InstallTitleBarLeftMenu NC.InstallCopyButtonEventFn NC.NoteFileTitleLeftWhenSelectedFn
NC.MakeNoteFileIconOperationsMenus NC.ResetCachedMenus NC.CreateNoteFileMenuItems
NC.GetNoteFileMenu NC.SetNoteFileMenu NC.PrintReadOnlyMsg NC.CopyMenu NC.ReadOnlyTEditSelfFn
NC.MakeTEditReadOnly NC.MakeTEditReadWrite NC.MakeMenusReadOnly NC.MakeMenusReadWrite)
```

;;; Selection Mechanism

```
(INITVARS (NC.SelectionInProgress NIL)
(NC.CardImageObjFns (IMAGEFNSCREATE)))
(FNS NC.SelectNoteCards NC.BinLoopProcess NC.ZapBinLoopProcess NC.SelectNoteCardsCopyInsertFn
NC.CopyButtonEventFn NC.ShrukenIconButtonEventFn NC.CardImageObjP NC.MakeCardImageObject)
(GLOBALVARS NC.CardImageObjFns)
```

;;; Fix up Lisp

```
[INITADVISE (LOGOUT (NIL (BEFORE NIL (if (NULL (NC.LogoutAdvice))
then
(RETURN)
```

;;; Stuff for creating and operating the NoteCards session icon.

```
(GLOBALVARS NC.CardOpsMenu NC.CardOpsItems NC.NoteFileOpsMenu NC.NoteFileOpsItems NC.OtherOpsMenu
NC.OtherOpsItems NC.NoteCardsIconBitmap)
(INITVARS (NC.NoteCardsIconWindow NIL))
```

## ;;; Bitmaps

```
(BITMAPS NC.NoteCardsIconBitmap NC.NewCardsIconBitmap NC.BoxesIconBitmap NC.NoteCardsIconMask
NC.NoteCardsIconShrunkenBitmap NC.NoteCardsIconShrunkenMask)
```

```
...
```

```
(FNS NC.NoteCardsWindowP NC.NoteCardsIconWindowP NC.MakeNoteCardsIcon NC.BringUpNoteCardsIcon
NC.NoteCardsIconAfterMoveFn NC.FileBrowserMenu NC.ListOfNoteFilesMenu NC.CheckpointSession
NC.CloseSession)
(ADDVARS (NC.NoteFileIconOperationsMenuItems (Open NIL "Opens this NoteFile." (SUBITEMS (
Open% Read/Write
NIL "Opens a
notefile for
Read/Write
access.")
(Open% Read-only
NIL "Open a
notefile for
read-only.))))
(Checkpoint NIL "Checkpoint this NoteFile, saving changed cards.")
(Close NIL "Closes this NoteFile." (SUBITEMS (Close% Notefile NIL "Close a notefile."
(|Close w/o confirm| NIL "Close this notefile
w/o asking for user confirm before
closing open cards.))))
(Abort NIL "Close NF, deleting changes since last checkpoint.")
(Compact NIL "Compacts this NoteFile to a new file." (SUBITEMS (|Compact To New File|
NIL "Compact a NoteFile
copying to a new file."
(|Compact In Place| NIL
"Compact a NoteFile
w/o copying to a new
file.))))
(Inspect&Repair NIL "Inspects and optionally repairs this NoteFile."
(SUBITEMS (From% Links NIL "Do an Inspect & Repair on a NoteFile from the Link
information.")
(From% Contents NIL "Inspects and optionally repairs a Notefile, but reads
every card substance. This slows it WAY down.))))
(Copy NIL "Copies this notefile to a target file.")
(Rename NIL "Rename this NoteFile")
>Delete NIL "Deletes this NoteFile.))
(NC.NoteFileIconOpenOperations Checkpoint Close Close% Notefile |Close w/o confirm| Abort)
(NC.NoteFileIconCloseOperations Open Open% Read/Write Open% Read-only Compact
|Compact To New File| |Compact In Place| Inspect&Repair From% Links From% Contents Copy
Rename Delete))
(GLOBALVARS NC.InitialNoteFileIconOperationsMenuItems NC.InitialNoteFileIconOpenOperations
NC.InitialNoteFileIconCloseOperations NC.ShowPropListMenu NC.EditPropListMenu)
(INITVARS (NC.InitialNoteFileIconOperationsMenuItems NC.NoteFileIconOperationsMenuItems)
(NC.InitialNoteFileIconOpenOperations NC.NoteFileIconOpenOperations)
(NC.InitialNoteFileIconCloseOperations NC.NoteFileIconCloseOperations))
[INITVARS (NC.DefaultMiddleButtonMenuItems NIL)
(NC.ShowPropListMenu (create MENU ITEMS _ ' ("Quit" (NC.ClosePropListEditor W 'NoSave)
"Quit from pointer list display.))
CENTERFLG _ T MENUFONT _ NC.MenuFont ITEMHEIGHT _
(IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
1)))
(NC.EditPropListMenu (create MENU ITEMS _ ' ("Add New Property" (NC.AddPropToPropList W)
"Add a new property to this card's property
list.")
("Delete Selected Property" (NC.DelPropFromList W)
"Delete selected property from this card's
property list.")
("Quit w/o Saving Changes" (NC.ClosePropListEditor
W
'NoSave)
"Quit from property list edit. Changes are not
saved.")
("Quit - Saving Changes" (NC.ClosePropListEditor
W
'Save)
"Quit from property list editor. Save
changes.))
CENTERFLG _ T MENUFONT _ NC.MenuFont ITEMHEIGHT _
(IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
1)))
[NC.LocalInsertLinkMenuItem ' (Insert% Link (FUNCTION NC.AddLinkToCard)
"Insert a link to another card at the currently selected
point in the text."
(SUBITEMS (Insert% Link (FUNCTION NC.AddLinkToCard)
"Insert a link to another card at the
currently selected point in the text."))
```

```

        (Insert% Links (FUNCTION NC.AddLinksToCard)
          "Insert links to other cards at the currently
            selected point in the text.")
        (|Add Global Link| (FUNCTION NC.AddGlobalLinkToCard)
          "Add a global link emanating from this card.")
        (|Add Global Links| (FUNCTION NC.AddGlobalLinksToCard)
          "Add multiple global links emanating from this
            card.")
[NC.GlobalInsertLinkMenuItem ' (|Add Global Link| (FUNCTION NC.AddGlobalLinkToCard)
  "Add a global link emanating from this card."
  (SUBITEMS (|Add Global Link| (FUNCTION
    NC.AddGlobalLinkToCard)
    "Add a global link emanating from this
      card.")
    (|Add Global Links| (FUNCTION NC.AddGlobalLinksToCard
      )
      "Add multiple global links emanating from this
        card.")
  (NC.DefaultLeftButtonMenuItems `((|Edit Property List| (FUNCTION NC.EditProperties)
    "Brings up an editor for the property list of this
      card.")
    (Show% Links (FUNCTION NC.ShowLinks)
      "Brings up a list of the links to and from this card.")
    (Show% Info (FUNCTION NC.ShowInfo)
      "Brings up a window containing special information about
        this card."
      (SUBITEMS (Indicate% NoteFile (FUNCTION
        NC.AttachNoteFileName
        )
        "Shows the name of this card's
          NoteFile.")))
    (Designate% FileBoxes (FUNCTION NC.AddParents)
      "File this note card in one or more file boxes."
      (SUBITEMS (|Unfile from FileBoxes| (FUNCTION
        NC.UnfileNoteCard
        )
        "Remove this card from one or more of
          its file boxes.")))
    (Assign% Title (FUNCTION NC.AssignTitle)
      "Assigns a (new) title to this note card."
      (Title/FileBoxes [FUNCTION (LAMBDA (TextStream)
        (NC.AssignTitle TextStream)
        (NC.AddParents TextStream)
        "Do both assigning of title and filing in fileboxes."
        ,NC.LocalInsertLinkMenuItem
        (|Close and Save| (FUNCTION CLOSEW)
          "Close this note card after saving it in the NoteFile."
          (SUBITEMS (|Close and Save| (FUNCTION CLOSEW)
            "Close this note card after saving it in
              the NoteFile."
            (|Close w/o Saving| (FUNCTION NC.QuitWithoutSaving
              )
              "Close this note card without saving any
                changes made since the last Save."
            (|Save w/o Closing| (FUNCTION NC.CardSaveFn)
              "Save this card in the NoteFile but don't
                close the card."
            (Delete% Card (FUNCTION NC.DeleteNoteCard)
              "Permanently delete this card from the
                NoteFile."
              (ADDVARS [NC.CardOpsItems (| Close | (NC.CloseNoteCards NIL NC.NoteCardsIconWindow)
                "Close several note cards."
                (SUBITEMS (Close% NoteCards (NC.CloseNoteCards NIL
                  NC.NoteCardsIconWindow)
                  "Close several note cards."
                  (Close% Structure (NC.CloseStructure NIL NIL
                    NC.NoteCardsIconWindow)
                    "Close note cards belonging to a structure.")))
                (| Delete | (NC.DeleteNoteCards NIL T NIL NC.NoteCardsIconWindow)
                  "Delete several note cards."
                  (SUBITEMS (Delete% NoteCards (NC.DeleteNoteCards NIL T NIL NC.NoteCardsIconWindow)
                    )
                    "Delete several note cards."
                    (DeleteStructure (NC.DeleteStructure NIL NIL NC.NoteCardsIconWindow)
                      "Delete note cards belonging to a structure.")))
                (| Copy | (NC.CopyCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
                  "Copy note cards to a filebox"
                  (SUBITEMS (Copy% Cards (NC.CopyCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
                    "Copy note cards to a filebox"
                    (Copy% Structure (NC.CopyStructure NIL NIL NIL NC.NoteCardsIconWindow)
                      "Copy note cards belonging to a structure.")))
                (| Move | (NC.MoveCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
                  "Move note cards to a filebox"
                  (SUBITEMS (Move% Cards (NC.MoveCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
                    "Move note cards to a filebox"
                    (Move% Structure (NC.MoveStructure NIL NIL NIL NC.NoteCardsIconWindow)
                      "Move note cards belonging to a structure."

```

```

(NC.NoteFileOpsItems (| Open | (NC.DoNoteFileOp 'Open% NoteFile)
    "Opens a notefile for Read/Write access."
    (SUBITEMS (Open% Read/Write (NC.DoNoteFileOp 'Open% NoteFile)
        "Opens a notefile for Read/Write access."
        (Open% Read-only (NC.DoNoteFileOp 'Read-only% Open)
            "Open a notefile for read-only.")))
(| Compact | (NC.DoNoteFileOp 'Compact% NoteFile)
    "Compact a NoteFile copying to a new file."
    (SUBITEMS (|Compact To New File| (NC.DoNoteFileOp 'Compact% NoteFile)
        "Compact a NoteFile copying to a new file."
        (|Compact In Place| (NC.DoNoteFileOp 'Compact In Place|)
            "Compact a NoteFile w/o copying to a new file.")))
(| Inspect&Repair | (NC.DoNoteFileOp 'Inspect&Repair% NoteFile)
    "Do an Inspect & Repair on a NoteFile from the Link information."
    (SUBITEMS (From% Links (NC.DoNoteFileOp 'NC.DoNoteFileOp)
        "Do an Inspect & Repair on a NoteFile from the Link information."
        (From% Contents (NC.DoNoteFileOp 'Read% Substances)
            "Inspects and optionally repairs a Notefile, but reads every
            substance. This slows it WAY down.")))
(| Copy | (NC.DoNoteFileOp 'Copy% NoteFile)
    "Copy a NoteFile."
(| Rename | (NC.DoNoteFileOp 'Rename% NoteFile)
    "Rename a NoteFile")
(| Delete | (NC.DoNoteFileOp 'Delete% NoteFile)
    "Delete a NoteFile")
(| Create | (NC.DoNoteFileOp 'Create% NoteFile)
    "Create a new NoteFile w/o opening it."
    (SUBITEMS (| Create without Open | (NC.DoNoteFileOp 'Create% NoteFile)
        "Create a new NoteFile w/o opening it."
        (| Create and Open | (NC.DoNoteFileOp 'Create and Open|)
            "Create a new NoteFile and open it.")))
(| ----- | NIL "")
(| Checkpoint | (NC.DoNoteFileOp 'Checkpoint% NoteFile)
    "Checkpoint a NoteFile"
    (SUBITEMS (Checkpoint% Notefile (NC.DoNoteFileOp 'Checkpoint% NoteFile)
        "Checkpoint a NoteFile."
        (Checkpoint% All (NC.CheckpointSession)
            "Checkpoint all currently open NoteFiles")))
(| Close | (NC.DoNoteFileOp 'Close% NoteFile)
    "Close a notefile."
    (SUBITEMS (Close% Notefile (NC.DoNoteFileOp 'Close% NoteFile)
        "Close a notefile."
        (|Close w/o confirm| (NC.DoNoteFileOp 'Close w/o confirm|)
            "Close a notefile w/o asking for user confirm before closing open
            cards."
        (Close% All (NC.CloseSession)
            "Close all currently open NoteFiles")
        (|Close All w/o confirm| (NC.CloseSession NIL T)
            "Close all open notefiles w/o asking for user confirm before closing
            open cards.")))
(| Abort | (NC.DoNoteFileOp 'Abort% NoteFile)
    "Close NF, deleting changes since last checkpoint."
    (SUBITEMS (Abort% Notefile (NC.DoNoteFileOp 'Abort% NoteFile)
        "Close Notefile, delete changes since last checkpoint."
        (Abort% All (NC.CloseSession T)
            "Abort all currently open NoteFiles")))
(| ----- | NIL "")
(| NC FileBrowser | (NC.FileBrowserMenu NC.NoteCardsIconWindow)
    "Bring up a NoteFile browser.")
(NC.OtherOpsItems (| Edit Parameters | (NC.BuildInspector)
    "Adjust NoteCards profile for this session."
(| NF Indicators On | (NC.ShowNoteFiles)
    "Turn on the NoteFile indicator for all currently open cards"
    (SUBITEMS (Indicators% On (NC.ShowNoteFiles)
        "Turn on the NoteFile indicator for all currently open cards")
        (Indicators% Off (NC.ShowNoteFiles T)
            "Turn off any NoteFile indicator on each card on the screen.")))
(| TEdit Killer On |
    (START-TEDIT-KILLER)
    "Start the TEDIT-KILLER process."
    (SUBITEMS (| TEdit Killer On | (START-TEDIT-KILLER)
        "Start the TEDIT-KILLER process."
        (|TEdit Killer Off| (STOP-TEDIT-KILLER)
            "Stop the TEDIT-KILLER process."
        (|New Process Limit| (SETQ TEDIT-PROCESS-LIMIT
            (OR (NUMBERP (RNUMBER (CONCAT "Old TEdit process limit:
" TEDIT-PROCESS-LIMIT)
                NIL NIL NIL T))
            TEDIT-PROCESS-LIMIT))
        "Change value of TEdit process limit."
        (|New Wait Time| (SETQ TEDIT-KILLER-WAIT-TIME
            (OR (LET* [(MULTIPLIER 100)
                (NEW-WAIT-TIME (NUMBERP
                    (RNUMBER (CONCAT "Current TEdit
                    killer wait
                    time: "
                    (IQUOTIENT

```

```

TEDIT-KILLER-WAIT-TIME
100)
" sec")
NIL NIL NIL T]
(AND NEW-WAIT-TIME (ITIMES NEW-WAIT-TIME
MULTIPLIER)))
TEDIT-KILLER-WAIT-TIME))
"Change value of Tedit killer wait time.)))
(| Load Patches | (NC.LoadPatches)
"Load the latest NoteCards patches.)))
(GLOBALVARS NC.InitialCardOpsItems NC.InitialNoteFileOpsItems NC.InitialOtherOpsItems)
(INITVARS (NC.InitialCardOpsItems NC.CardOpsItems)
(NC.InitialNoteFileOpsItems NC.NoteFileOpsItems)
(NC.InitialOtherOpsItems NC.OtherOpsItems))
(INITVARS (NC.CardOpsMenu)
(NC.OtherOpsMenu)
(NC.NoteFileOpsMenu))
(GLOBALVARS NC.NoteCardsIconMenuBitMap NC.OldStyleIconMenuBitmap NC.OldStyleIconBitmap
NC.OldStyleIconMenuBitmap NC.NoteCardsIconMask NC.NoteCardsIconShrunkenBitmap
NC.NoteCardsIconShrunkenMask NC.OtherOpsItems NC.OtherOpsMenu)
(INITVARS (NC.NoteCardsIconMenuBitMap)
(NC.OldStyleIconMenuBitmap))
(FNS NC.RedisplayNoteCardsIconWindow NC.NoteCardsIconButtonEventFn NC.DoNoteFileOp
NC.NoteCardsIconCursorEventFn NC.NoteCardsIconFn NC.ResetNoteCardsIcon NC.MakeNewStyleNoteCardsIcon
NC.MakeOldStyleNoteCardsIcon)

```

### ;;; Command Menu interface to NC functions

```
(FNS NC.CloseNoteCards NC.CopyStructure NC.CloseStructure NC.DeleteStructure NC.MoveStructure)
```

### ;;; NoteFile interface

```
(FNS NC.NoteFileMenuWindowP NC.SetUpNoteFileInterface NC.ResetNoteFileInterface
NC.NoteFileIconButtonEventFn NC.NoteFileOperations NC.NoteFileIconMiddleButtonFn
NC.PopUpNoteFileName NC.DisplayFileCapacity)
[INITVARS (NC.DefaultNoteFileIconMiddleButtonItem '(|Full File Name| NC.PopUpNoteFileName "Display
full file name of this notefile.")
(File% Capacity NC.DisplayFileCapacity "Display
file capacity of this notefile")]
```

### ;;; Show NoteFile Indicators

```
(FNS NC.ShowNoteFiles NC.AttachNoteFileName NC.MakeNoteFileNameIndicator NC.NoteFileIndicatorWinHeight)
```

### ;;; File Browser

```
(GLOBALVARS NC.FileBrowserMenuItemsToRemove NC.FileBrowserMenuItemsToAdd NC.FileBrowserMenuItems
NC.FileBrowserPatterns NC.FileBrowserDefaultPatterns)
[INITVARS (NC.FileBrowserMenuItemsToRemove ' (Hardcopy See Edit Load Compile Copy Rename Expunge))
(NC.FileBrowserMenuItemsToKeep ' (Delete Undelete Recompute Sort))
[NC.FileBrowserMenuItemsToAdd ' ((Open NC.FileBrowserOpen "Open selected Notefiles."
(SUBITEMS (Open% Read/Write NC.FileBrowserOpen "Opens a
notefile for Read/Write access.")
(Read-only% Open (NC.FileBrowserOpen INPUT)
"Open selected Notefiles for read-only.)))
(Checkpoint NC.FileBrowserCheckpoint "Checkpoint the selected
Notefiles, saving dirty cards.")
(Close NC.FileBrowserClose "Close selected Notefiles.")
(Abort NC.FileBrowserAbort "Aborts the selected Notefiles losing
work since last checkpoint.")
(Compact NC.FileBrowserCompact "Compacts selected Notefiles
copying to new files." (SUBITEMS
(|Compact To New File|
NC.FileBrowserCompact
"Compacts selected
Notefiles copying to new
files.")
(|Compact In Place|
(NC.FileBrowserCompact
' InPlace)
"Compacts selected Notefiles w/o
copyig to new files.)))
(Inspect&Repair NC.FileBrowserInspect&Repair "Inspects and
optionally repairs selected Notefiles."
(SUBITEMS (From% Links NC.FileBrowserInspect&Repair "Do an
Inspect & Repair on a NoteFile from the
Link information.")
(From% Contents (NC.FileBrowserInspect&Repair
'ReadSubstances)
"Inspects and optionally repairs selected
Notefiles, but reads every substance. This
slows things WAY down.)))
(Copy FB.COPYCOMMAND "Copies given notefile(s) to target file (or

```

```

        directory).")
        (Rename FB.RENAMECOMMAND "Moves given notefile(s) to target file
        (or directory).")
(NC.FileBrowserExpungeMenuItem ' (Expunge NC.FileBrowserExpunge "Permanently removes from the file
        system all files marked for deletion"))
(NC.FileBrowserRecomputeMenuItem ' (Recompute FB.UPDATECOMMAND "Recomputes set of notefiles
        satisfying selection pattern" (SUBITEMS ("Same Pattern"
                                                    FB.UPDATECOMMAND
                                                    "Recomputes set
                                                    of files
                                                    satisfying
                                                    current
                                                    pattern")
                                                    ("New Pattern"
NC.FileBrowserRecomputeWithNewPattern
        "Prompts for a new
        selection pattern
        and updates
        browser")
        ("New Info"
        FB.NEWINFOCOMMAND
        "Change the
        set of file
        attributes
        that are
        displayed"]

```

;;; Build the menu items for notecards file browser.

```

[INITVARS (NC.FileBrowserMenuItems (APPEND NC.FileBrowserMenuItemsToAdd
        (for MenuItem in FB.MENU.ITEMS when (FMEMB (CAR MenuItem)
                                                    NC.FileBrowserMenuItemsToKeep
                                                    )
        collect
        (if (EQ (CAR MenuItem)
                'Recompute)
            then NC.FileBrowserRecomputeMenuItem else MenuItem))
        (LIST NC.FileBrowserExpungeMenuItem)
        (for MenuItem in FB.MENU.ITEMS unless
            (OR (FMEMB (CAR MenuItem)
                    NC.FileBrowserMenuItemsToRemove)
                (FMEMB (CAR MenuItem)
                    NC.FileBrowserMenuItemsToKeep))
            collect MenuItem]
[INITVARS (NC.FileBrowserPatterns NIL)
        (NC.FileBrowserDefaultPatterns ' ({DSK}/HOME/IL/NOTECARDS/NOTEFILES/*.NOTEFILE;* {DSK}*.NOTEFILE;*
        )

```

;;; Functions called from file browser menu.

```

(FNS NC.FileBrowserOpen NC.FileBrowserClose NC.FileBrowserCheckpoint NC.FileBrowserAbort
        NC.FileBrowserCompact NC.FileBrowserInspect&Repair NC.FileBrowserExpunge NC.GetFileBrowserPattern
        NC.FileBrowserRecomputeWithNewPattern NC.FindFileBrowserWinForPattern)
(FNS NC.FlashFileBrowserWin)
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
        NCINTERFACE)))

```

;;; Internal variables

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

```

(GLOBALVARS NC.NoticedNoteFilesMenuNewItem NC.NoticedNoteFilesMenu NC.CachedMenus NC.NoteCardsIconPosition
        NC.SelectionInProgress NC.SelectedCards NC.SelectingSourcesMenu NC.SelectingContentsMenu
        NC.SelectingSingleCardMenu NC.SelectingMultipleCardsMenu NC.DeleteSelectingMenu NC.SelectingParentsMenu
        NC.SelectingBrowserSourceMenu NC.SelectingFileBoxChildrenMenu NC.SelectingCardsMenu NC.SelectingCardMenu
        NC.TEditMenus NC.NoteCardTypeMenu NC.DatabaseOperationsMenu NC.LinkLabelMenu NC.WindowRightButtonMenu
        NC.LogoWindow LOGOW ExecWindow PROCESS.STATUS.WINDOW QUADTREE.HIGHRES NC.BrowseOrSearchMenu
        FONTDIRECTORIES NC.DatabaseFileNameSuggestion NC.UncachingNotCompleted NC.UnspecifiedLinkLabel
        NC.UCASESystemLinkLabels NC.MainMenuPosition NC.NoteCardsIconWindow NC.NewCardsIconBitmap
        NC.BoxesIconBitmap NC.NoteFileIndicatorFont NC.NoteFileMenuLingerFlg NC.FileBrowserExpungeMenuItem
        NC.FileBrowserRecomputeMenuItem NC.FileBrowserMenuItemsToKeep NC.DefaultNoteFileIconMiddleButtonItem
        NC.OpenedNoteFileMenu NC.ClosedNoteFileMenu NC.NoteFileIconOperationsMenuItems
        NC.NoteFileIconOpenOperations NC.NoteFileIconCloseOperations NC.NoteCardTypeMenu
        NC.DefaultLeftButtonItemMenuItems NC.LocalInsertLinkMenuItem NC.GlobalInsertLinkMenuItem
        NC.DefaultMiddleButtonItemMenuItems NC.PromptWindowMonitorLock)
)

```

```
(RPAQ? NC.LinkLabelMenu NIL)
```

```
(RPAQ? NC.LinkDisplayModesMenu NIL)
```

```

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

(RPAQ? NC.DatabaseFileNameSuggestion NIL)

(RPAQ? NC.TEditMenus NIL)

(RPAQ? NC.OpenedNoteFileMenu NIL)

(RPAQ? NC.ClosedNoteFileMenu NIL)

(RPAQ? NC.NoteCardsIconPosition NIL)

(RPAQ? NC.PromptWindowMonitorLock (CREATE.MONITORLOCK "PromptWindowMonitorLock"))

(RPAQ? NC.FileNameFont (FONTCREATE 'HELVETICA 10 'BOLD))

(RPAQ? NC.FullFileNameFont (FONTCREATE 'HELVETICA 8 'STANDARD))

(RPAQ? NC.MenuTitleFont (FONTCREATE 'HELVETICA 10 'BOLD))

(RPAQ? NC.MaxFileNameChars 18)

;;; Setup and top level interaction

(DECLARE%: DONTEVAL@LOAD DONTCOPY EVAL@COMPILE

(LOAD? (NC.FindFile 'FILEBROWSER NIL NIL T 'LISPUSERSDIRECTORIES)
'PROP)
)

(DEFINEQ

NoteCards
[LAMBDA (IconPosition)
(* fgh%: " 7-Jun-86 16:21")
(* * Start up NoteCards by bringing up the NoteCards icon at IconPosition)
(* * fgh |6/7/86| First created.)
(NC.BringUpNoteCardsIcon (if (POSITIONP IconPosition)
then IconPosition])

NC.MakeMainMenu
[LAMBDA (DatabaseTitle)
; Edited 17-Jan-89 17:36 by rtk

;;; Create the NoteCards menus

;;; rht 8/1/84: Added def'n for new menu: NC.SelectingFileBoxChildrenMenu

;;; rht 2/4/85: The browser source menu now accepts multiple selections.

;;; fkr 10/29/85: Now moves window onto screen.

;;; fgh 11/14/85 Eliminated NC Main Menu which was replaced by per NoteFile menus.

;;; pmi 12/4/86 Added menus for left button title bar of NoteFile Icons, one for open NoteFile and one for closed NoteFile

;;; pmi 12/19/86 Removed menu items 'Compact' and 'Inspect&Repair' from NC.OpenedNoteFileMenu.

;;; rht 2/11/87: Moved stuff for creating NC.OpenedNoteFileMenu and NC.ClosedNoteFileMenu out of here and into a separate function.

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

;;; bk 1/17/89: Use global NC.MenuTitleFont for title font

(DECLARE (GLOBALVARS NC.MenuFont NC.MainMenuPosition NC.SelectingSourcesMenu NC.SelectingContentsMenu
NC.SelectingSingleCardMenu NC.SelectingCardMenu NC.SelectingMultipleCardsMenu
NC.DeleteSelectingMenu NC.SelectingParentsMenu NC.SelectingBrowserSourceMenu
NC.SelectingFileBoxChildrenMenu NC.SelectingCardsMenu NC.MenuTitleFont WHITESHADE GRAYSHADE
))
(PROG [X (TitleFont NC.MenuTitleFont)
(Position (OR (POSITIONP NC.MainMenuPosition)
(create POSITION
XCOORD _ 350
YCOORD _ 650]
(* Main Menu)

(SETQ X
(create MENU
ITEMS _ '((NoSource NIL "Indicates that this note card has no source.")
(Undo NIL "Backup over last selection.")
(Done NIL "Indicates that selection is completed.))
WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
TITLE _ " Selecting Source(s) "
CENTERFLG _ T
MENUBORDERSIZE _ 1
MENUOUTLINESIZE _ 1
MENUCOLUMNS _ 3
MENUFONT _ NC.MenuFont

```

```

ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
MENUPOSITION _ Position
MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingSourcesMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((Cancel NIL "Cancel card closing.")
                (Undo NIL "Backup over last selection.")
                (Done NIL "Indicates that selection is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting File Box(es) "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingContentsMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((New% Card NIL "Make a new note card.")
                (Cancel NIL "Cancel this selection.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting Note Card "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 2
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingSingleCardMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((Cancel NIL "Cancel this selection.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting Single Card "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 1
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingCardMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((Cancel NIL "Cancel this selection.")
                (Undo NIL "Backup over last selection.")
                (Done NIL "Indicates that selection is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting Note Cards "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingMultipleCardsMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((Cancel NIL "Cancel this selection.")
                (Undo NIL "Backup over last selection.")
                (Done NIL "Indicates that selection is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting cards to close or delete "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.DeleteSelectingMenu X)
(SETQ X
  (create MENU
    ITEMS _ ' ((Cancel NIL "Cancel this selection.")
                (Undo NIL "Backup over last selection.")
                (Done NIL "Indicates that selection is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)

```



```

TITLE _ " Selecting FileBox(es) "
CENTERFLG _ T
MENUBORDERSIZE _ 1
MENUOUTLINESIZE _ 1
MENCOLUMNS _ 3
MENUFONT _ NC.MenuFont
ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
MENUPOSITION _ Position
MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingParentsMenu X)
(SETQ X
  (create MENU
    ITEMS _ '((Cancel NIL "Cancel this selection.")
              (Undo NIL "Backup over last selection.")
              (Done NIL "Indicates that this operation is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting Starting Card(s)/Box(es) "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingBrowserSourceMenu X)
(SETQ X
  (create MENU
    ITEMS _ '((Cancel NIL "Cancel this operation.")
              (Undo NIL "Backup over last selection.")
              (Done NIL "Indicates that this operation is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting cards to file "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingFileBoxChildrenMenu X)
(SETQ X
  (create MENU
    ITEMS _ '((Cancel NIL "Cancel this operation.")
              (Undo NIL "Backup over last selection.")
              (Done NIL "Indicates that this operation is completed.)))
    WHENSELECTEDFN _ (FUNCTION NC.SelectionMenusWhenSelectedFn)
    TITLE _ " Selecting Cards "
    CENTERFLG _ T
    MENUBORDERSIZE _ 1
    MENUOUTLINESIZE _ 1
    MENCOLUMNS _ 3
    MENUFONT _ NC.MenuFont
    ITEMHEIGHT _ (IPLUS 10 (FONTPROP NC.MenuFont 'HEIGHT))
    MENUPOSITION _ Position
    MENUTITLEFONT _ TitleFont))
(SETQ NC.SelectingCardsMenu X])

```

**(NC.LogoutAdvice**

[LAMBDA NIL

```

; Edited 6-Sep-88 17:35 by pmi
; Advice attached to logout to prevent logging out with an open
; database.

```

```

;;; fgh 11/14/85 Updated to handle Multiple open noteFiles.

```

```

;;; rht 11/16/85: Modified to call NC.ListOfNoteFiles.

```

```

;;; pmi 6/9/88: Fixes problem where logging out with notefiles opened caused a break for Illegal Stack Arg 'LOGOUT.  Removed (RETFROM 'LOGOUT)
;;; at end of advice because RETFROM no longer works the same way in Lyric.  Instead, this now returns T if it's okay for LOGOUT to proceed, and NIL
;;; if it should abort.  This will be checked in the actual advice added to LOGOUT.

```

```

(LET ((OpenNoteFiles (for NoteFile in (NC.ListOfNoteFiles) when (AND (STREAMP (fetch (NoteFile Stream)
                                                                    of NoteFile))
                              (OPENP (fetch (NoteFile Stream)
                                              of NoteFile))))
      collect NoteFile)))
  (if OpenNoteFiles
    then (NC.PrintMsg NIL T "The following NoteFiles are still open." (CHARACTER 13)
      "They must be closed before you can logout!!!"
      (CHARACTER 13))
    (for NF in OpenNoteFiles do (NC.PrintMsg NIL NIL (fetch (NoteFile FullFileName) of NF)
      ", ")))
  (RPTQ 1 (PROGN (VIDEOCOLOR T)
    (DISMISS 100)
    (VIDEOCOLOR NIL))

```

```

                (DISMISS 100)))
        (RPTQ 3 (FLASHW PROMPTWINDOW))
        NIL
    else T])

```

**(NC.AttachPromptWindow**

[LAMBDA (MainWindow MinWidth Height Font DontCreate)

; Edited 8-Dec-88 11:22 by krivacic

;; GETPROMPTWINDOW that works for small windows. sizes default to 300 X 60

;; fgh 5/1/86 MainWindowXSize now based on MainWindow's REGION WIDTH not its REGION RIGHT.

;; fgh 6/6/86 To fix many bugs made into a wrapper for NC.GETPROMPTWINDOW

;; rht 3/21/87: Now keeps MinWidth small enough so that the prompt window will always fit on the screen without moving MainWindow.

```

(DECLARE (GLOBALVARS NC.PrintMsgFont))
(if MainWindow
  then (if MinWidth
    then (SETQ MinWidth (MIN MinWidth (NC.MaxPromptWindowWidth MainWindow)))
    else (SETQ MinWidth 300))
  (OR Height (SETQ Height 60))
  (NC.GETPROMPTWINDOW MainWindow (IQUOTIENT Height (if (FONTP Font)
    then (FONTHEIGHT Font)
    else 15))
    (OR Font NC.PrintMsgFont)
    DontCreate MinWidth))

```

**(NC.AttachPromptWindowOffScreen**

[LAMBDA (MainWindow MinWidth Height Font)

(\* rht%: "17-Apr-87 16:06")

(\* Moves MainWindow off screen, calls NC.AttachPromptWindow, closes resulting promptwindow, and moves window back.)

```

(DECLARE (GLOBALVARS NC.OffScreenPosition))
(LET ((OldPos (WINDOWPOSITION MainWindow))
  PromptWin)
  (MOVEW MainWindow NC.OffScreenPosition)
  (SETQ PromptWin (NC.AttachPromptWindow MainWindow MinWidth Height Font))
  (AND (OPENWP PromptWin)
    (CLOSEW PromptWin))
  (MOVEW MainWindow OldPos))

```

**(NC.GETPROMPTWINDOW**

[LAMBDA (MAINWINDOW %#LINES FONT DONTCREATE MINWIDTH)

; Edited 12-Dec-88 10:07 by krivacic

;; makes sure that MAINWINDOW has an attached promptwindow and returns it. If one already exists, it is shaped to be at least #LINES high. If FONT is NIL, the font of the main window is used for the promptwindow.

;; fgh 6/6/86 Adapted from system GETPROMPTWINDOW. Added MINWIDTH arg.

;; rht 7/2/86: Now attaches prompt window on right if near right edge of screen.

;; rht 7/4/86: Yanked out PositionOnEdge localvar. Put code in place.

;; rht 3/21/87: AttachWindowPositionOnEdge to attach prompt window is left if possible, then center, then right. Also doesn't let prompt window be wider than screen.

;; rht 3/30/87: More ATTACHEDWINDOWS hacking. This time to get the unstacking and restacking of attachedwindows right during reshaping of the prompt window.

;; rht&amp;rg&amp;pmi 4/22/87: Added stack of extra prompt windows and checks for whether they have processes equal to this one.

;; rht 5/25/88: The CREATEW call now uses off-screen position instead of 0,0.

```

(DECLARE (GLOBALVARS WBorder NC.OffScreenPosition))
(AND MINWIDTH (SETQ MINWIDTH (MIN (DIFFERENCE SCREENWIDTH (TIMES WBorder 2))
  MINWIDTH)))
(WITH.MONITOR NC.PromptWindowMonitorLock
  (PROG ((PWINDOWPROP (WINDOWPROP MAINWINDOW 'PROMPTWINDOW))
    (PWINDOWMINWIDTH (WINDOWPROP MAINWINDOW 'PROMPTWINDOWMINWIDTH))
    (MAINWINDOWREGION (WINDOWPROP MAINWINDOW 'REGION))
    (OtherPWindowProps (WINDOWPROP MAINWINDOW 'OtherPromptWindows))
    PWINDOW PWINDOWREGION WIDTH HEIGHT OBSCUREDHEIGHT PWINDOW.PWINDOW PWinWidth
    AttachWindowPositionOnEdge ThisPWindowProp)
    (SETQ PWinWidth (WIDTHIFWINDOW (OR MINWIDTH PWINDOWMINWIDTH 1)))
    [SETQ AttachWindowPositionOnEdge (COND
      ((LEQ (PLUS (fetch (REGION LEFT) of MAINWINDOWREGION)
        PWinWidth)
        SCREENWIDTH)
        'LEFT)
      ((LEQ (PLUS (fetch (REGION LEFT) of MAINWINDOWREGION)
        (LRSH (PLUS PWinWidth (fetch (REGION WIDTH)
          of MAINWINDOWREGION))
          1))
        SCREENWIDTH)
        'CENTER)
      (T 'RIGHT)
    ))
    ; Find a free prompt window.

```

```

[for PWindowProp in (CONS PWINDOWPROP OtherPWindowProps)
do (LET [(PWindowProcess (WINDOWPROP (CAR PWindowProp)
                                     'PromptWindowProcess)]
      (if (EQ (THIS.PROCESS)
              PWindowProcess)
          then (SETQ ThisPWindowProp PWindowProp)
              (RETURN PWindowProp))
      (if (AND (NULL ThisPWindowProp)
                (NOT (PROCESSP PWindowProcess)))
          then (SETQ ThisPWindowProp PWindowProp)

[COND
  (DONTCREATE (RETURN (CAR ThisPWindowProp)))
  [ThisPWindowProp
   (SETQ PWINDOW (CAR ThisPWindowProp))
   (WINDOWPROP PWINDOW 'PromptWindowProcess (THIS.PROCESS))
   (if (SETQ PWINDOW.PWINDOW (WINDOWPROP PWINDOW 'PROMPTWINDOW))
       then
           ; This prompt window has a prompt window of its own. Close
           ; and detach it.

           (REMOVEWINDOW (CAR PWINDOW.PWINDOW)))
   (WINDOWPROP PWINDOW 'MAXSIZE (CONS 64000 64000))
   (WINDOWPROP PWINDOW 'MINSIZE (CONS 1 1))
   [SETQ HEIGHT (HEIGHTIFWINDOW (TIMES (OR %#LINES (CDR ThisPWindowProp)
                                           1)
                                     (FONTPROP (DSPFONT NIL PWINDOW)
                                                  'HEIGHT]
   (SETQ WIDTH (MAX PwinWidth (fetch (REGION WIDTH) of MAINWINDOWREGION)))
   (SETQ PWINDOWREGION (WINDOWPROP PWINDOW 'REGION))
   [COND
     ((NOT (OPENWP PWINDOW))
      (ATTACHWINDOW PWINDOW MAINWINDOW 'TOP AttachWindowPositionOnEdge 'LOCALCLOSE]
   (COND
     ([OR (NOT (EQP HEIGHT (fetch (REGION HEIGHT) of PWINDOWREGION)))
          (NOT (EQP WIDTH (fetch (REGION WIDTH) of PWINDOWREGION]
          ;; Window exists, but not right size. Keep windows detached during reshaping in case AttachWindowPositionOnEdge has
          ;; changed.

```

;;; Need to distinguish among the windows attached to MAINWINDOW between those attached before PWINDOW and those attached after. The latter  
 ;;; need to be reattached at AttachWindowPositionOnEdge after PWINDOW is reshaped.

```

[LET [(DescriptionList (for Win in (ATTACHEDWINDOWS MAINWINDOW) bind WA
                              when [EQ 'TOP (CAR (SETQ WA (WINDOWPROP Win 'WHEREATTACHED]
                              collect
                                  ; Detach attached windows at the top of the main Win and return
                                  ; a list describing their attachment.
                                  (PROG1 (LIST Win (CDR WA)
                                              (WINDOWPROP Win 'PASSTOMAINCOMS))
                                         (DETACHWINDOW Win]
                                  ; Have to smash these to let SHAPEW work.
                                  (WINDOWPROP PWINDOW 'MAXSIZE NIL)
                                  (WINDOWPROP PWINDOW 'MINSIZE NIL)
                                  (WINDOWPROP PWINDOW 'PASSTOMAINCOMS NIL)
                                  (RESETLST
                                   [RESETSAVE NIL `(PROGN (WINDOWPROP ,PWINDOW 'OPENFN
                                                             ', (WINDOWPROP PWINDOW 'OPENFN NIL]
                                   (SHAPEW PWINDOW (create REGION using PWINDOWREGION HEIGHT _ HEIGHT WIDTH _ WIDTH)))

```

;;; Now reattach all the windows in order to their old locations, except that PWINDOW and those after it should be attached at  
 ;;; AttachWindowPositionOnEdge.

```

(for WindowDescription in DescriptionList bind SeenPWINDOW
do (LET ((Win (CAR WindowDescription))
      (if (EQ Win PWINDOW)
          then (SETQ SeenPWINDOW T)
          (ATTACHWINDOW Win MAINWINDOW 'TOP (if SeenPWINDOW
                                                  then AttachWindowPositionOnEdge
                                                  else (CADDR WindowDescription)))
      (WINDOWPROP Win 'PASSTOMAINCOMS (CADDR WindowDescription]
(RPLACD ThisPWindowProp %#LINES) ; Fall through to check visibility

]
(T (SETQ PWINDOW (CREATEW [create REGION
                              LEFT _ (fetch (POSITION XCOORD) of NC.OffScreenPosition)
                              BOTTOM _ (fetch (POSITION YCOORD) of NC.OffScreenPosition)
                              WIDTH _ (MAX PwinWidth (fetch (REGION WIDTH) of MAINWINDOWREGION
                                                              ))
                              HEIGHT _ (SETQ HEIGHT
                                              (HEIGHTIFWINDOW (TIMES (OR %#LINES (SETQ %#LINES 1)
                                                                      (FONTPROP
                                                                       (OR FONT (SETQ FONT
                                                                           (DSPFONT NIL
                                                                           MAINWINDOW)
                                                                           'HEIGHT]
                                                                      ))
                                              (DSPSCROLL T PWINDOW)
                                              (DSPFONT FONT PWINDOW)
                                              (WINDOWPROP PWINDOW 'PAGEFULLFN 'NILL)
                                              (ATTACHWINDOW PWINDOW MAINWINDOW 'TOP AttachWindowPositionOnEdge 'LOCALCLOSE)

```

```

    (if (NULL PWINDOWPROP)
      then (WINDOWPROP MAINWINDOW 'PROMPTWINDOW (CONS PWINDOW %#LINES))
      else (WINDOWADDPROP MAINWINDOW 'OtherPromptWindows (CONS PWINDOW %#LINES)))
    (WINDOWPROP PWINDOW 'PromptWindowProcess (THIS.PROCESS))
    (WINDOWPROP PWINDOW 'OPENFN (FUNCTION \PROMPTWINDOW.OPENFN)
[COND
  ((ILESSP [SETQ OBSCUREDHEIGHT (IDIFFERENCE SCREENHEIGHT (fetch (REGION TOP)
                                                                of (WINDOWPROP PWINDOW 'REGION]
                                                                0)
                                                                ; Promptwindow off screen at top, so slip window group down to
                                                                ; make it visible
    (RELMOVEW MAINWINDOW (create POSITION
                                  XCOORD _ 0
                                  YCOORD _ OBSCUREDHEIGHT]
    (WINDOWPROP PWINDOW 'MINSIZE (CONS 1 HEIGHT))
    (WINDOWPROP PWINDOW 'MAXSIZE (CONS 64000 HEIGHT))
    (WINDOWPROP MAINWINDOW 'PROMPTWINDOWMINWIDTH (OR MINWIDTH 1))
    (OPENW PWINDOW)
    (RETURN PWINDOW))))))

```

**(NC.MaxPromptWindowWidth**

[LAMBDA (MainWindow)

(\* rht%: "23-Mar-87 11:26")

(\* Returns the maximum interior width that a prompt window for MainWindow could have and still be on the screen and attached, no matter whether it's positioned left, right or center.)

```

(DECLARE (GLOBALVARS WBorder))
(LET ((WindowRegion (WINDOWPROP MainWindow 'REGION))
      Left RightSpace Right Width)
  (SETQ Width (fetch (REGION WIDTH) of WindowRegion))
  (DIFFERENCE (if [LESSP (SETQ Left (fetch (REGION LEFT) of WindowRegion))
                     (SETQ RightSpace (DIFFERENCE SCREENWIDTH (SETQ Right (PLUS Left Width)
                                         then (MAX (DIFFERENCE SCREENWIDTH Left)
                                         (PLUS (TIMES 2 Left)
                                         Width))
                     else (MAX Right (PLUS (TIMES 2 RightSpace)
                                         Width)))
    (TIMES 2 WBorder]))

```

**(NC.PromptWindowOpenP**

[LAMBDA (MainWindow)

(\* rht%: " 8-Jun-87 11:43")

(\* Return non-nil if MainWindow has an open prompt window.)

```

(OR [OPENWP (CAR (WINDOWPROP MainWindow 'PROMPTWINDOW]
  (for PromptWinPair in (WINDOWPROP MainWindow 'OtherPromptWindows) thereis (OPENWP (CAR PromptWinPair))
)

```

;;; Menus of all sorts

```

(RPAQ? NC.CachedMenus ' (NC.NoteCardTypeMenu NC.LinkIconMiddleButtonMenu NC.CardOpsMenu NC.NoteFileOpsMenu
                        NC.OtherOpsMenu))

```

```

(RPAQ? NC.DesiredTypeOrdering ' (Text FileBox Sketch Graph BAR Browser Search LinkIndex Document))

```

```

(RPAQ? NC.NoticedNoteFilesMenu )

```

```

(RPAQ? NC.NoticedNoteFilesMenuNewItem )

```

```

(DEFINEQ

```

**(NC.AskLinkLabel**

```

[LAMBDA (MainWindow MultipleFlg SystemLinksFlg NewLinkFlg CancelOkayFlg ReverseLinksFlg OldLinkLabels
      ReturnListOfListFlg)
; Edited 30-May-88 16:41 by Trigg
; Asks for label on notecard links

```

;; rht 8/2/84: Added double columns for when called by the browser, i.e. ReverseLinksFlg=T. Reverse links have prefix '\_ '.

;; rht 11/19/84: Changed strings from 'pointer' to 'link' and from 'label' to 'type'.

;; rht 2/14/85: Added extra arg OldLinkLabels in the Multiple selection case to display previous choices.

;; rht 10/11/85: Took out printing to prompt window. It's a waste of screen space.

;; rht 10/22/85: Added ReturnListOfListFlg so caller can tell difference between aborting from Stylesheet and choosing NULL set of links.

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

;; pmi 12/5/86: Modified so that clicking outside of Link type menu is equivalent to choosing \*\*CANCEL\*\* from menu.

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

;; rht 5/30/88: Link typess menu now comes up at cursorpos rather than in corner of MainWindow.

```

(DECLARE (GLOBALVARS NC.MenuFont NC.UCASESystemLinkLabels NC.UnspecifiedLinkLabel))
(PROG (Menu Choice Choices LabelsList LinkLabels Position Card NoteFile)
  (SETQ Card (NC.CoerceToCard MainWindow))
  (SETQ NoteFile (fetch (Card NoteFile) of Card))
  (SETQ LinkLabels (NC.RetrieveLinkLabels NoteFile SystemLinksFlg))

```

```

[SETQ Position (AND (WINDOWP MainWindow)
  (create POSITION
    XCOORD _ (fetch (REGION LEFT) of (WINDOWPROP MainWindow 'REGION))
    YCOORD _ (fetch (REGION TOP) of (WINDOWREGION MainWindow))
  )
)
[COND
  (MultipleFlg [SETQ Choices (STYLESHEET
    (CREATE.STYLE 'ITEMS
      [COND
        [ReverseLinksFlg
          (LIST (create MENU
            ITEMS _ LinkLabels)
          (create MENU
            ITEMS _ (for Link in LinkLabels
              collect (PACK* '_ Link]
          (T (LIST (create MENU
            ITEMS _ LinkLabels]
          'NEED.NOT.FILL.IN
          'MULTI
          'POSITION Position 'TITLE "Link Types?" 'SELECTIONS
          (LIST (for Label in OldLinkLabels
            when (NEQ (NTHCHAR Label 1)
              '_)
            collect Label)
          (for Label in OldLinkLabels
            when (EQ (NTHCHAR Label 1)
              '_)
            collect Label]
          )
        )
      )
    )
  )
  (RETURN (COND
    ((NULL Choices) ; User aborted from stylesheet.
      NIL)
    [(AND ReverseLinksFlg ReturnListOfListFlg)
      (LIST (NCONC (CAR Choices)
        (CADR Choices)
        (ReverseLinksFlg (NCONC (CAR Choices)
          (CADR Choices)))
        (ReturnListOfListFlg Choices)
        (T (CAR Choices)
          (SETQ Menu (create MENU
            TITLE _ " Link Type "
            ITEMS _ [NCONC (COND
              [ReverseLinksFlg (for Link in LinkLabels
                join (LIST Link (CONCAT "_" Link]
                (T (COPY LinkLabels)))
                (AND NewLinkFlg (LIST ' |--New Link Type--|))
                (AND CancelOkayFlg (LIST '**CANCEL**])
              )
            )
            CENTERFLG _ T
            MENUCOLUMNS _ (AND ReverseLinksFlg 2)
            MENUFONT _ NC.MenuFont
            ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
              1)))
          )
        ]
      )
    ]
  )
)

```

;;; Allow user to cancel by selecting outside of Links menu

```

(SETQ Choice (OR (MKATOM (MENU Menu))
  '**CANCEL**))
[COND
  ((EQ Choice '**CANCEL**))
  (SETQ Choice))
  ((EQ Choice ' |--New Link Type--|))
  (NC.ClearMsg MainWindow NIL)
  (until (COND
    ((NOT (FMEMB [U-CASE (SETQ Choice (MKATOM (NC.AskUser "Enter new link type: " NIL NIL NIL
      MainWindow NIL]
      NC.UCASESystemLinkLabels))
    T)
    (T (NC.PrintMsg MainWindow T Choice " is a system reserved link type." (CHARACTER 13)
      "Please choose another link type."
      (CHARACTER 13))
      NIL)))
    (OR Choice (SETQ Choice NC.UnspecifiedLinkLabel))
    (AND (NOT (FMEMB Choice LinkLabels))
      (NC.StoreLinkLabels NoteFile (CONS Choice LinkLabels]
    )
  )
  (RETURN Choice]
)

```

## (NC.AskNoteCardType

[LAMBDA (MainMenuOrRegion)

; Edited 8-Dec-88 12:26 by krivacic  
; Ask user to choose a note card type

;; rht 1/12/85: Now takes an optional Region argument dictating where to place the NoteCardTypeMenu. If NIL, then uses MainMenu.

;; fgh 11/16/85 Updated to take a MainMenu arg in place of using the NC.MainMenu globalvar.

;; pmi 3/20/87: Changed fields of NC.NoteCardTypeMenu: CENTERFLG to T, MENUFONT to NC.MenuFont, changed title from 'Type?' to 'Card Types'

;; pmi 12/31/87: Changed ManinMenuOrRegion to MainMenuOrRegion.

;; rht 2/22/88: Now allows NIL MainMenuOrRegion arg which brings up menu at cursor position.

```

(DECLARE (GLOBALVARS NC.NoteCardTypeMenu NC.MenuFont))
(PROG (W Z CardType)
  (OR (AND (BOUNDP 'NC.NoteCardTypeMenu)
    (type? MENU NC.NoteCardTypeMenu))
    (SETQ NC.NoteCardTypeMenu (create MENU
      ITEMS _ (NC.ExpandBars (NC.SortCardTypes (NC.ListOfCardTypes T)))
      CENTERFLG _ T
      TITLE _ "Card Types"
      MENUFONT _ NC.MenuFont
      ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
        1)
      MENCOLUMNS _ 1
      CHANGEOFFSETFLG _ T)))
  [AND NIL (replace MENUPOSITION of NC.NoteCardTypeMenu
    with (COND
      [(REGIONP MainMenuOrRegion)
        (CONS (fetch (REGION LEFT) of MainMenuOrRegion)
          (IPLUS (fetch (REGION BOTTOM) of MainMenuOrRegion)
            (fetch (REGION HEIGHT) of MainMenuOrRegion)
            (IMINUS (fetch (MENU IMAGEHEIGHT) of NC.NoteCardTypeMenu)
              1))
          (type? MENU MainMenuOrRegion)
          (CONS [IPLUS (fetch (REGION LEFT) of (SETQ Z (WINDOWPROP (WFROMMENU
            'REGION]
            (fetch (REGION LEFT) of (SETQ W
              (MENUITEMREGION
                (CAR (NTH (fetch (MENU ITEMS) of
                  MainMenuOrRegion)
                    1))
                MainMenuOrRegion]
              (IPLUS (fetch (REGION BOTTOM) of Z)
                (fetch (REGION TOP) of W)
                (IMINUS (fetch (MENU IMAGEHEIGHT) of NC.NoteCardTypeMenu)
                  1))
              (T NIL]
            (SETQ CardType (MENU NC.NoteCardTypeMenu))
            (if (STRPOS "---" CardType)
              then (SETQ CardType NIL))
            (RETURN CardType])

```

**(NC.SortCardTypes**

[LAMBDA (CARD-TYPE-LIST)

; Edited 12-Jan-89 11:20 by rtk

;; Change The Sort order of the card types by the desired type ordering

```

(LET ((SYSTEM-CARD-LIST NIL)
  (USER-CARD-LIST NIL)
  (SAW-BAR NIL))

```

;; First sort the desired notecard types into system or user created cards

```

(for INDEX from 0 as CARD-TYPE in NC.DesiredTypeOrdering when (OR (EQ 'BAR CARD-TYPE)
  (FMEMB CARD-TYPE CARD-TYPE-LIST))
  do (SETQ SAW-BAR (OR SAW-BAR (EQ 'BAR CARD-TYPE)))
    [if (OR SAW-BAR (GETPROP CARD-TYPE 'SYSTEM-GENERATED))
      then [if (NOT (EQ 'BAR CARD-TYPE))
        then (PUTPROP CARD-TYPE 'SYSTEM-GENERATED T)
        (SETQ SYSTEM-CARD-LIST (APPEND SYSTEM-CARD-LIST (LIST CARD-TYPE]
      else (SETQ USER-CARD-LIST (APPEND USER-CARD-LIST (LIST CARD-TYPE]
        CARD-TYPE)

```

;; Add any Card Types Not in the Desired Ordering List

```

[AND (for CARD-TYPE in CARD-TYPE-LIST when (NOT (OR (FMEMB CARD-TYPE USER-CARD-LIST)
  (FMEMB CARD-TYPE SYSTEM-CARD-LIST)))
  do ;; determine where to add the card
    (if (GETPROP CARD-TYPE 'SYSTEM-GENERATED)
      then ;; system generated card type, so it goes in the system list
        (SETQ SYSTEM-CARD-LIST (APPEND SYSTEM-CARD-LIST (LIST CARD-TYPE)))
      else (SETQ USER-CARD-LIST (APPEND USER-CARD-LIST (LIST CARD-TYPE]

```

;; Return the resulting card list

```

(APPEND USER-CARD-LIST (LIST 'BAR)
  SYSTEM-CARD-LIST])

```

**(NC.ExpandBars**

[LAMBDA (SORTED-ITEM-LIST)

; Edited 15-Dec-88 11:46 by krivacic

;; Replace BAR in the SORTED-ITEM-LIST with aa string of "---", which is equal in character length to the longest name in SORTED-ITEM-LIST

```

(LET* ((LAST-MAX 0)
  BAR-LINE)
  [MAPCAR SORTED-ITEM-LIST (FUNCTION (LAMBDA (X)
    (SETQ LAST-MAX (MAX (NCHARS (OR (AND (LISTP X)
      (CAR X))
        X))
    LAST-MAX]

```



;;; rht 3/23/87: Now looks for PromptWindow on SelectionPromptWindow instead of looking for main window.

;;; rg 4/1/87 changed CANCELLED to DON'T

```
(LET [(PromptWindow (WINDOWPROP (WFROMMENU Menu)
                                'SelectionPromptWindow))
      (SELECTQ (CAR Item)
                ((Done)
                 [WINDOWPROP PromptWindow 'SelectedCards (CONS 'DONE (WINDOWPROP PromptWindow 'SelectedCards))
                 ((Abort Cancel)
                  (WINDOWPROP PromptWindow 'SelectedCards 'DON'T))
                 (New% Card [WINDOWPROP PromptWindow 'SelectedCards (CONS '*New% Card* (WINDOWPROP PromptWindow
                                                                                          'SelectedCards))
                 (NoSource (WINDOWPROP PromptWindow 'SelectedCards (LIST 'NC00000)))
                 (Undo [WINDOWPROP PromptWindow 'SelectedCards (CONS '*Undo% Selection* (WINDOWPROP PromptWindow
                                                                                          'SelectedCards))
                NIL)
      (NOTIFY.EVENT (WINDOWPROP PromptWindow 'CopyInsertEvent))
```

### (NC.InstallTitleBarButtonEventFn

[LAMBDA (Window TitleBarButtonEventFn)

; Edited 7-Oct-2021 13:17 by medley

(\* \* Save the old BUTTONEVENTFN and replace with ours.)

(\* \* fgh 2021-06-09 Added support for the MODERNIZE lispusers package which alters how the left button handling work within the title bar.)

```
(OR (EQ (WINDOWPROP Window 'BUTTONEVENTFN)
        TitleBarButtonEventFn)
    (WINDOWPROP Window 'OLDBUTTONEVENTFN (WINDOWPROP Window 'BUTTONEVENTFN TitleBarButtonEventFn)))
(AND (GETD 'MODERNWINDOW)
     (MODERNWINDOW Window NIL NC.ModernizeTITLEPROPORTION))
```

### (NC.InstallTitleBarLeftMenu

[LAMBDA (Window CardType)

; Edited 7-Oct-2021 15:33 by medley

(\* \* Make a menu from the CardType's LeftButtonMenuItems and install on windowprop.)

(\* \* pmi 3/25/87%: Added NC.MenuFont to all menus)

```
(DECLARE (GLOBALVARS NC.MenuFont))
(WINDOWPROP Window 'TitleBarLeftButtonMenu (create MENU
                                                    ITEMS _ (NC.GetCardTypeField LeftButtonMenuItems CardType)
                                                    CENTERFLG _ T
                                                    MENUFONT _ NC.MenuFont
                                                    ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                                                    1))
```

### (NC.InstallCopyButtonEventFn

[LAMBDA (Window)

(\* rht%: "13-May-87 10:52")

(\* \* Save the current COPYBUTTONEVENTFN and stash ours.)

```
(OR (EQ (WINDOWPROP Window 'COPYBUTTONEVENTFN)
        (FUNCTION NC.CopyButtonEventFn))
    (WINDOWPROP Window 'OldCopyButtonEventFn (WINDOWPROP Window 'COPYBUTTONEVENTFN (FUNCTION
                                                                                          NC.CopyButtonEventFn))
```

### (NC.NoteFileTitleLeftWhenSelectedFn

[LAMBDA (Item Menu Button)

; Edited 23-Feb-89 11:01 by krivacic

;;; WhenSelectedFn for the menus NC.OpenedNoteFileMenu and NC.ClosedNoteFileMenu. These operations used to be performed by NC.NoteFileOperations. Provides faster Left-button title bar menu on NoteFile icons.

;;; rht 2/11/87: Now handles possibly extra unanticipated menu items.

;;; rht 2/11/87: Removed useless call to DEFAULTWHENSELECTEDFN.

;;; rht 2/16/87: Added Close cards w/o confirm case.

;;; pmi 5/14/87: Removed HashArray argument to NC.OpenNoteFile.

;;; pmi 5/27/87: Changed 'Compact to Target File' to 'Compact to New File.'

;;; bk 12/22/87 add (OR (CADR |Item|) (CAR |Item|)) for Re-named menu items.

;;; bk 12/26/89: Removed (OR (CADR |Item|) (CAR |Item|)) for operation, since menu items were not renamed & library cards used the apply fn. instead, make sure the item is in the "other" notefileoperations menu before rejecting the operation.

;;; bk 2/23/89: Added Bringup TOC parm to NC.OpenNoteFile

```
(DECLARE (GLOBALVARS NC.NoteFileIconCloseOperations NC.NoteFileIconOpenOperations NC.MsgDelay NC.BringUpTOC
```



```

))
(LET ((NoteFile (GETMENUPROP Menu 'NoteFile))
      (Operation (CAR Item))
      NoteFileMenuWindow)
  (SETQ NoteFileMenuWindow (WFROMMENU (fetch (NoteFile Menu) of NoteFile)))
  (WINDOWPROP NoteFileMenuWindow 'BusyOperation Operation)
  (COND
    ((AND (NC.NoteFileOpenP NoteFile)
          (NOT (FMEMB Operation NC.NoteFileIconOpenOperations))
          (FMEMB Operation NC.NoteFileIconCloseOperations))
      (NC.PrintMsg NoteFileMenuWindow T "Can't " Operation " an open notefile." (CHARACTER 13))
      (DISMISS NC.MsgDelay)
      (NC.ClearMsg NoteFileMenuWindow T))
    ((AND (NOT (NC.NoteFileOpenP NoteFile))
          (NOT (FMEMB Operation NC.NoteFileIconCloseOperations))
          (FMEMB Operation NC.NoteFileIconOpenOperations))
      (NC.PrintMsg NoteFileMenuWindow T "Can't " Operation " a closed notefile." (CHARACTER 13))
      (DISMISS NC.MsgDelay)
      (NC.ClearMsg NoteFileMenuWindow T))
    (T (SELECTQ Operation
      (Open (NC.OpenNoteFile NoteFile NIL NIL NIL NIL NIL NIL NoteFileMenuWindow NIL NIL NIL
        NIL NIL NIL NC.BringUpTOC))
      (Open% Read/Write
        (NC.OpenNoteFile NoteFile NIL NIL NIL NIL NIL NIL NoteFileMenuWindow NIL NIL NIL NIL
        NIL NIL NC.BringUpTOC))
      (Open% Read-only
        (NC.OpenNoteFile NoteFile NIL NIL NIL NIL NIL NIL NoteFileMenuWindow NIL NIL NIL T
        NIL NIL NC.BringUpTOC))
      (Checkpoint (NC.CheckpointNoteFile NoteFile NIL NIL NoteFileMenuWindow))
      (Close (NC.CloseNoteFile NoteFile NoteFileMenuWindow))
      (Close% Notefile
        (NC.CloseNoteFile NoteFile NoteFileMenuWindow))
      (|Close w/o confirm|
        (NC.CloseNoteFile NoteFile NoteFileMenuWindow T))
      (Abort (NC.AbortSession NoteFile NoteFileMenuWindow))
      (Compact (NC.CompactNoteFile NoteFile NIL NIL NoteFileMenuWindow))
      (|Compact To New File|
        (NC.CompactNoteFile NoteFile NIL NIL NoteFileMenuWindow))
      (|Compact In Place|
        (NC.CompactNoteFile NoteFile NIL T NoteFileMenuWindow))
      (Inspect&Repair
        (NC.InspectAndRepairNoteFile NoteFile NIL NoteFileMenuWindow))
      (From% Links (NC.InspectAndRepairNoteFile NoteFile NIL NoteFileMenuWindow))
      (From% Contents
        (NC.InspectAndRepairNoteFile NoteFile T NoteFileMenuWindow))
      (Copy (NC.CopyNoteFile NoteFile NIL NoteFileMenuWindow))
      (Rename (NC.RenameNoteFile NoteFile))
      (Delete (NC.DeleteDatabaseFile NoteFile NoteFileMenuWindow))
      (APPLY* (CADR Item)
        NoteFile NoteFileMenuWindow]))

```

**(NC.MakeNoteFileIconOperationsMenus**

[LAMBDA NIL

; Edited 15-Dec-88 10:22 by krivacic

;; Rebuild the menus NC.OpenedNoteFileMenu and NC.ClosedNoteFileMenu smashing any current values.

;; pmi 9/12/88: Now uses NCP.GrayShade instead of GRAYSHADE to shade the disabled menu items.

```

(DECLARE (GLOBALVARS NC.OpenedNoteFileMenu NC.ClosedNoteFileMenu NC.NoteFileIconOperationsMenuItems
  NC.MenuFont NC.NoteFileIconOpenOperations NC.NoteFileIconCloseOperations NCP.GrayShade))
; First, make two identical menus.

```

```

(SETQ NC.OpenedNoteFileMenu (create MENU
  ITEMS _ (for Item in NC.NoteFileIconOperationsMenuItems
    when (MEMB (CAR Item)
      NC.NoteFileIconOpenOperations)
    collect Item)
  CHANGEOFFSETFLG _ 'Y
  MENUOFFSET _ (CONS -1 0)
  CENTERFLG _ NIL
  TITLE _ "NoteFile Ops"
  MENUTITLEFONT _ NC.MenuFont
  MENUFONT _ NC.MenuFont
  ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
    1)
  WHENSELECTEDFN _ (FUNCTION NC.NoteFileTitleLeftWhenSelectedFn)))
(SETQ NC.ClosedNoteFileMenu (create MENU
  ITEMS _ (for Item in NC.NoteFileIconOperationsMenuItems
    when (MEMB (CAR Item)
      NC.NoteFileIconCloseOperations)
    collect Item)
  CHANGEOFFSETFLG _ 'Y
  MENUOFFSET _ (CONS -1 0)
  CENTERFLG _ NIL
  TITLE _ "NoteFile Ops"
  MENUTITLEFONT _ NC.MenuFont
  MENUFONT _ NC.MenuFont
  ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
    1)

```

```
(PUTPROP FullFileName 'OpenMenuItem (LIST OpenMenuBitmap ',FullFileName (CONCAT "Selects NoteFile "  
FullFileName)))
```

```

(PUTPROP FullFileName 'OpenMenuItemShaded (LIST OpenMenuBitmapShaded '',FullFileName (CONCAT
"Selects
NoteFile "
FullFileName
)))

```

;;; Create bitmaps for the closed notefile, shaded and unshaded.

```

(SETQ ClosedMenuBitmap (BITMAPCREATE BitmapWidth BitmapHeight))
(SETQ ClosedMenuBitmapShaded (BITMAPCREATE BitmapWidth BitmapHeight))
(SETQ ClosedDisplayStream (DSPCREATE ClosedMenuBitmap))
(SETQ ClosedDisplayStreamShaded (DSPCREATE ClosedMenuBitmapShaded))
(MOVETO (PLUS SpaceWidth oWidth SpaceWidth)
FontDescent ClosedDisplayStream)
(DSPFONT NC.FileNameFont ClosedDisplayStream)
(PRIN1 FileNameString ClosedDisplayStream)
(MOVETO FullNameMargin FontDescent ClosedDisplayStream)
(DSPFONT NC.FullFileNameFont ClosedDisplayStream)
(PRIN1 FullFileNameString ClosedDisplayStream)
(BITBLT ClosedMenuBitmap 0 0 ClosedMenuBitmapShaded 0 0)
(BLTSHADE NCP.GrayShade ClosedDisplayStreamShaded (PLUS SpaceWidth oWidth)
0
(PLUS SpaceWidth NameWidth SpaceWidth)
BitmapHeight
'ERASE)

```

;;; Construct the full menu items for a closed notefile.

```

(PUTPROP FullFileName 'ClosedMenuItem (LIST ClosedMenuBitmap '',FullFileName (CONCAT "Selects NoteFile
" FullFileName)))
(PUTPROP FullFileName 'ClosedMenuItemShaded (LIST ClosedMenuBitmapShaded '',FullFileName
(CONCAT "Selects NoteFile " FullFileName)))
(RETURN FullFileName))

```

### (NC.GetNoteFileMenu

[LAMBDA (NoteFileOrFileName)

; Edited 7-Dec-88 11:02 by krivacic

;;; pmi 6/2/87: Created to get the NoteFile icon menu for a NoteFile.

```

(LET (FullFileName NoteFile)
  (if (type? NoteFile NoteFileOrFileName)
    then (OR (fetch (NoteFile Menu) of NoteFileOrFileName)
      (GETPROP (fetch (NoteFile FullFileName)
        NoteFileOrFileName)
        'Menu))
    elseif (SETQ FullFileName (FULLNAME NoteFileOrFileName))
    then (OR (GETPROP FullFileName 'Menu)
      (if (type? NoteFile (SETQ NoteFile (NC.NoteFileFromFileName FullFileName)))
        then (fetch (NoteFile Menu) of NoteFile])))

```

### (NC.SetNoteFileMenu

[LAMBDA (NoteFileOrFileName Menu)

; Edited 7-Dec-88 11:01 by krivacic

;;; pmi 6/2/87: Created to set the NoteFile icon menu for a NoteFile.

```

(LET (FullFileName NoteFile)
  (if NoteFileOrFileName
    then (if (type? NoteFile NoteFileOrFileName)
      then (replace (NoteFile Menu) of NoteFileOrFileName with Menu)
      (PUTPROP (fetch (NoteFile FullFileName)
        NoteFileOrFileName)
        'Menu Menu)
    elseif (SETQ FullFileName (FULLNAME NoteFileOrFileName))
    then (PUTPROP FullFileName 'Menu)
      (if (type? NoteFile (SETQ NoteFile (NC.NoteFileFromFileName FullFileName)))
        then (replace (NoteFile Menu) of NoteFile with Menu)))

```

### (NC.PrintReadOnlyMsg

[LAMBDA (Stream)

(\* Randy.Gobbel "22-Oct-87 14:17")

(\* print "not allowed on read-only cards" msg. Code copied from NC.PrintOperationInProgressMsg)

(\* rg [10/21/87] First created.)

```

(LET* [(Window (NC.FetchWindow (NC.CoerceToCard Stream)))
(MainWindowRegion (WINDOWPROP Window 'REGION))
(PWinWidth (MAX (WIDTHIFWINDOW (fetch (REGION WIDTH) of MainWindowRegion))
300))
(PWindow (OR (for WCandidate in (ALLATTACHEDWINDOWS Window) thereis (AND (WINDOWPROP WCandidate
'NC.ReadOnlyMsgWindow)
(OPENWP WCandidate))))
(CREATEW [create REGION
LEFT _ 0

```

(NC.CopyMenu

(NC.ReadOnlyTEditSelfFn

(NC.MakeTEditReadOnly

[illegible]

(NC.MakeTEditReadWrite

(\* Randy.Gobbel "22-Oct-87 14:28")

## ; Edited 27-Jan-88 16:20 by MacDonald

```
[LAMBDA (Window AddSafeItems AddSafeSubItems) ; Edited 27-Jan-88 16:20 by MacDonald
(LET*
  ((TitleBarReadOnlyLeftMenu (WINDOWPROP Window 'TITLEBARREADONLYLEFTMENU))
   (TitleBarReadOnlyMiddleMenu (WINDOWPROP Window 'TITLEBARREADONLYMIDDLEMENU))
   (SafeItems AddSafeItems)
   (SafeSubItems AddSafeSubItems)
   NewItems SubItems NewSubItems)
;; If Window doesn't have RO AND RW Menus then create them. The RW Menus are just the menus present on Open; the RO Menus are
;; modifications of these.
[if (NOT (OR TitleBarReadOnlyLeftMenu TitleBarReadOnlyMiddleMenu))
  then
    [WINDOWPROP Window 'TITLEBARREADWRITELEFTMENU (NC.CopyMenu (WINDOWPROP Window 'TitleBarLeftButtonMenu)
    [WINDOWPROP Window 'TITLEBARREADWRITEMIDDLEMENU (NC.CopyMenu (WINDOWPROP Window
                                                                'TitleBarMiddleButtonMenu)
    (SETQ TitleBarReadOnlyLeftMenu (WINDOWPROP Window 'TitleBarLeftButtonMenu))
    (SETQ TitleBarReadOnlyMiddleMenu (WINDOWPROP Window 'TitleBarMiddleButtonMenu))
    (for Menu in (LIST TitleBarReadOnlyLeftMenu TitleBarReadOnlyMiddleMenu)
      do
        (if Menu
          then
            (SETQ NewItems
              (for Item in (fetch (MENU ITEMS) of Menu) bind ItemName
                collect
                  (SETQ ItemName (if (LISTP Item)
                                     then (CAR Item)
                                     else Item)))
              (if (NOT (MEMB ItemName SafeItems))
                then [SETQ Item (LIST ItemName ' (FUNCTION NC.PrintReadOnlyMsg)
                                     (if (LISTP Item)
```

```

                                then (CADDR Item]
      else (SETQ SubItems (APPLY* (OR (fetch SUBITEMFN of Menu)
                                      (FUNCTION DEFAULTSUBITEMFN))
                                Menu Item))
        (if SubItems
          then (SETQ NewSubItems
                (for SubItem in SubItems
                  collect (SETQ ItemName (if (LISTP SubItem)
                                             then (CAR SubItem)
                                             else SubItem))
                        [if (NOT (MEMB ItemName SafeSubItems))
                          then (SETQ SubItem (LIST ItemName ' (FUNCTION NC.PrintReadOnlyMsg
                                                             )
                                                             (if (LISTP SubItem)
                                                             then (CADDR SubItem]
                                                             SubItem))
                        (RPLACD (CAR (NTH Item 4))
                              NewSubItems)))
          Item))
      (replace (MENU ITEMS) of Menu with NewItems)
      [for Item in (fetch (MENU ITEMS) of Menu)
      do
        (if (NOT (MEMB (if (LISTP Item)
                           then (CAR Item)
                           else Item)
                       SafeItems))
          then (SHADEITEM Item Menu GRAYSHADE)
          else (SETQ SubItems (APPLY* (OR (fetch SUBITEMFN of Menu)
                                          (FUNCTION DEFAULTSUBITEMFN))
                                      Menu Item))
            (if SubItems
              then (for SubItem in SubItems
                    do (if (NOT (MEMB (if (LISTP SubItem)
                                          then (CAR SubItem)
                                          else SubItem)
                                      SafeSubItems))
                        then (SHADEITEM SubItem Menu GRAYSHADE]
              (replace (MENU IMAGE) of Menu with NIL]
      (WINDOWPROP Window 'TitleBarLeftButtonMenu TitleBarReadOnlyLeftMenu)
      (WINDOWPROP Window 'TitleBarMiddleButtonMenu TitleBarReadOnlyMiddleMenu)
      (WINDOWPROP Window 'TITLEBARREADONLYLEFTMENU TitleBarReadOnlyLeftMenu)
      (WINDOWPROP Window 'TITLEBARREADONLYMIDDLEMENU TitleBarReadOnlyMiddleMenu])

```

**(NC.MakeMenusReadWrite**

```

[LAMBDA (Window) ; Edited 28-Jan-88 07:40 by MacDonald
  (LET* [(TitleBarReadOnlyMiddleMenu (WINDOWPROP Window 'TITLEBARREADONLYMIDDLEMENU))
        (TitleBarReadOnlyLeftMenu (WINDOWPROP Window 'TITLEBARREADONLYLEFTMENU))
        (TitleBarReadWriteLeftMenu (WINDOWPROP Window 'TITLEBARREADWRITELEFTMENU))
        (TitleBarReadWriteMiddleMenu (WINDOWPROP Window 'TITLEBARREADWRITEMIDDLEMENU))
    ;; Window doesn't have RO Menus then it's RO status has never been changed so we make the RW Menus from the original menus.
    [if (NOT (OR TitleBarReadOnlyMiddleMenu TitleBarReadOnlyLeftMenu))
      then [WINDOWPROP Window 'TITLEBARREADWRITELEFTMENU ' (WINDOWPROP Window 'TitleBarLeftButtonMenu]
            [WINDOWPROP Window 'TITLEBARREADWRITEMIDDLEMENU ' (WINDOWPROP Window
                                                                    'TitleBarMiddleButtonMenu]
            [SETQ TitleBarReadWriteLeftMenu (NC.CopyMenu (WINDOWPROP Window 'TITLEBARREADWRITELEFTMENU))
              (SETQ TitleBarReadWriteMiddleMenu (NC.CopyMenu (WINDOWPROP Window
                                                                'TITLEBARREADWRITEMIDDLEMENU))
              (WINDOWPROP Window 'TitleBarLeftButtonMenu TitleBarReadWriteLeftMenu)
              (WINDOWPROP Window 'TitleBarMiddleButtonMenu TitleBarReadWriteMiddleMenu])
  )

```

```

;;; Selection Mechanism

```

```

(RPAQ? NC.SelectionInProgress NIL)

```

```

(RPAQ? NC.CardImageObjFns (IMAGEFNSCREATE))

```

```

(DEFINEQ

```

**(NC.SelectNoteCards**

```

[LAMBDA (SingleCardFlg SelectionPredicate Menu InstigatingCardOrWindow Msg CheckForCancelFlg FileLevelLockFlg)
  ; Edited 11-Jan-89 15:38 by rtk

```

```

;;; Select a set of note cards or a single note card, depending on SingleCardFlg. Works by interpreting all mouse presses until a card has been chosen
;;; (if SingleCardFlg is T) or until the Done button has been pressed (if SingleCardFlg is NIL). If the mouse press occurs within a Title bar of a notecard,
;;; add that note card to the selected list. Otherwise, if you are pointing into a note card, call the BUTTONEVENTFN for that note card. The Selection in
;;; Progress flag has been set, so all note card BUTTONEVENTFNs should know to ignore all presses except those that occur on link icons. Link icon
;;; presses should simply add the destination of that link to the selected note cards list. This function should always be called from inside of an
;;; NC.CardSelectionOperation wrapper.

```

```

;; RG 4/1/87 changed CANCELLED to DON'T

```

```

;; rg 4/22/87 changed some names,

```

```

;; rht&rg&pmi 4/22/87: Moved location of ALLOW.BUTTON.EVENTS.

```

```

;; rg 6/2/87 added FileLevelLockFlg
;; rht 6/6/87: If user selects a cross-file link card, then try to follow it.
;; bk 1/10/89 Stop Multiple Links & text from being shift selected when shift selecting from text card to insert link .
(DECLARE (USEDFREE CardListResetVar))
;; if we are running under the mouse process, start up a new mouse process
(ALLOW.BUTTON.EVENTS)
(LET
  ((FirstTime T)
   SelectedCard Window Card ButtonEventFn InstigatingWindow InstigatingCard InstigatingNoteFile MenuWindow
   PromptWindow CopyInsertEvent CardProcessedEvent SelectedCards BinLoopProcess OldTTYProcess OpInProgress
   ResetItems TTYResetVar InternalResetVar)
  (NAMED-RESETLST
   InternalResetVar
   (OR SelectionPredicate (SETQ SelectionPredicate (FUNCTION TRUE)))
   (SETQ PromptWindow (OR (NC.AttachPromptWindow (SETQ InstigatingWindow (NC.CoerceToInterestedWindow
                                                         InstigatingCardOrWindow)))
                          PROMPTWINDOW))
   (SETQ InstigatingCard (NC.CoerceToCard InstigatingCardOrWindow))
   (SETQ InstigatingNoteFile (AND InstigatingCard (fetch (Card NoteFile) of InstigatingCard)))
   (NC.PrintMsg InstigatingWindow T (COND
                                     (Msg (CONCAT Msg (CHARACTER 13)))
                                     (T ""))
    "Items shift-selected: ")
   (SETQ OldTTYProcess (TTY.PROCESS))
   ;; Set up the prompt window for proper use by the CopyInsertFn
   (NAMED-RESETSAVE InternalResetVar (WINDOWPROP PromptWindow 'SelectCardsMonitor (CREATE.MONITORLOCK
                                                                                               'SelectCards))
    '(WINDOWPROP ,PromptWindow SelectCardsMonitor NIL))
   [WINDOWPROP PromptWindow 'CopyInsertEvent (SETQ CopyInsertEvent (CREATE.EVENT 'CopyInsertEvent])
   (WINDOWPROP PromptWindow 'NewCardsProcessed T)
   [WINDOWPROP PromptWindow 'CardProcessedEvent (SETQ CardProcessedEvent (CREATE.EVENT 'CardProcessedEvent])
   (WINDOWPROP PromptWindow 'SelectNoteCardsProcess (THIS.PROCESS))
   (WINDOWPROP PromptWindow 'COPYINSERTFN (FUNCTION NC.SelectNoteCardsCopyInsertFn))
   (NAMED-RESETSAVE InternalResetVar (WINDOWPROP PromptWindow 'SelectedCards NIL)
    '(WINDOWPROP ,PromptWindow SelectedCards NIL))
   (NAMED-RESETSAVE InternalResetVar (WINDOWPROP PromptWindow 'SelectingCards T)
    '(WINDOWPROP ,PromptWindow SelectingCards NIL))
   ;; make sure ^E aborts properly
   (NAMED-RESETSAVE InternalResetVar (RESET.INTERRUPTS (LISPINTERRUPTS)
                                                         T))
   ;; Make the process behind the prompt window including control for a blinking cursor
   [WINDOWPROP PromptWindow 'PROCESS (SETQ BinLoopProcess (ADD.PROCESS '(NC.BinLoopProcess)
                                                                 'WINDOW PromptWindow 'NAME 'BinLoopProcess
                                                                 'TTYENTRYFN
                                                                 [FUNCTION (LAMBDA (Process)
                                                                 (PROCESSPROP Process
                                                                 'OldCaret
                                                                 (CARET CROSSHAIRS))
                                                                 (ECHOMODE])
                                                                 'TTYEXITFN
                                                                 (FUNCTION (LAMBDA (Process)
                                                                 (CARET (PROCESSPROP Process
                                                                 'OldCaret))
                                                                 (ECHOMODE T])
                                                                 (NAMED-RESETSAVE InternalResetVar NIL '(DEL.PROCESS ,BinLoopProcess))
                                                                 ;; Insure the prompt window is cleared on the way out
                                                                 (NAMED-RESETSAVE InternalResetVar NIL '(PROGN (AND (HASTTYWINDOWP ,BinLoopProcess)
                                                                 (TTY.PROCESS (if (AND (PROCESSP ,OldTTYProcess)
                                                                 (HASTTYWINDOWP ,OldTTYProcess))
                                                                 then ,OldTTYProcess
                                                                 else T)))
                                                                 (NC.ClearMsg ,InstigatingWindow T])
                                                                 (WINDOWADDPROP PromptWindow 'CLOSEFN (FUNCTION NC.ZapBinLoopProcess))
                                                                 ;; Set up the menu above the prompt window
                                                                 ; fix in case MENUPOSITION is set incorrectly in menu passed
                                                                 ; down
                                                                 (replace (MENU MENUPOSITION) of Menu with (CONSTANT (create POSITION
                                                                 XCOORD _ 0
                                                                 YCOORD _ 0)))
                                                                 (NAMED-RESETSAVE InternalResetVar (PROGN (ATTACHMENU Menu (OR InstigatingWindow PROMPTWINDOW)
                                                                 (if InstigatingWindow
                                                                 then 'TOP
                                                                 else 'BOTTOM)
                                                                 (if (AND (WINDOWP InstigatingWindow)
                                                                 (WINDOWP PromptWindow))
                                                                 then (CDR (WINDOWPROP PromptWindow 'WHEREATTACHED))
                                                                 else 'LEFT))
                                                                 (WINDOWPROP (WFROMMENU Menu)
                                                                 'SelectionPromptWindow PromptWindow))
                                                                 '(PROGN (REMOVEWINDOW (WFROMMENU ,Menu)

```

```

;; If there is an instigatingWindow, make sure it and all its attachments are visible on the screen.
(if InstigatingWindow
  then (NC.MoveWindowOntoScreen InstigatingWindow))

;; Give the prompt window the tty process
(TTY.PROCESS (WINDOWPROP PromptWindow 'PROCESS))

;; Loop as long as necessary
(WITH.MONITOR (WINDOWPROP PromptWindow 'SelectCardsMonitor)
  [until (OR (EQ SelectedCards 'DON'T)
    [AND SingleCardFlg SelectedCards (NOT (WINDOWPROP PromptWindow 'ShiftSelectingCards)
      (EQ (CAR SelectedCards)
        'DONE))
    do (
      ; Wait for the user to respond by copy inserting something into
      ; the prompt window
      (until (OR (NOT (PROCESSP BinLoopProcess))
        (NOT (EQ SelectedCards (WINDOWPROP PromptWindow 'SelectedCards)
          do (MONITOR.AWAIT.EVENT (WINDOWPROP PromptWindow 'SelectCardsMonitor)
            CopyInsertEvent 1000))
            ; Get the latest selection list
            (SETQ SelectedCards (WINDOWPROP PromptWindow 'SelectedCards))
            (WINDOWPROP PromptWindow 'NewCardsProcessed T)
            (NOTIFY.EVENT CardProcessedEvent T)
            (OR (PROCESSP BinLoopProcess)
              (SETQ SelectedCards 'DON'T))
            (NAMED-RESETLST
              TTYResetVar
              (NAMED-RESETSAVE TTYResetVar (TTY.PROCESS (THIS.PROCESS)))
              ; Turn off the caret
              ; If the last thing wasn't a done or cancel, process the new
              ; selection
              (SETQ Card (CAR SelectedCards))
              (WITH.MONITOR NC.LockLock
                [COND
                  ((AND (NEQ Card 'DONE)
                    (NEQ SelectedCards 'DON'T)
                    (NEQ Card '*New% Card*))
                    ; Try to follow cross-file links.
                    [if (AND (NC.ValidCardP Card)
                      (NC.CrossFileLinkCardP Card))
                      then (SETQ Card (NC.GetCrossFileLinkDestCard Card InstigatingWindow))
                        (WINDOWPROP PromptWindow 'SelectedCards (SETQ SelectedCards
                          (CONS Card (CDR SelectedCards)
                            ; Check to make sure that the selection is valid
                            (COND
                              ((EQ Card '*Undo% Selection*)
                                ; Chop off two elements from the list --- the indicator and the
                                ; previous item.
                                (SETQ Card (CADR SelectedCards))
                                (WINDOWPROP PromptWindow 'SelectedCards (SETQ SelectedCards (CDDR SelectedCards)))
                                ; now get our hands off of all the locks we've acquired for this
                                ; card
                                (if Card
                                  then (NAMED-RESETUNSAVE CardListResetVar (NC.FetchUserDataProp Card
                                    'ResetItems))
                                    (NC.SetUserDataProp Card 'ResetItems NIL))
                                    (NC.ClearMsg InstigatingWindow NIL))
                                  [OR (NOT (NC.ValidCardP Card))
                                    (NULL (APPLY* SelectionPredicate Card)))
                                  ; Does this card match the selection predicate
                                  (NC.PrintMsg InstigatingWindow T "Invalid selection. ***" (CHARACTER 13))
                                  (WINDOWPROP PromptWindow 'SelectedCards (SETQ SelectedCards (CDR SelectedCards)
                                    (AND (SETQ OpInProgress (if FileLevelLockFlg
                                      then (NC.NoteFileCheckOpInProgress (fetch (Card
                                        NoteFile
                                          )
                                          of Card))
                                      else (NC.CardCheckOpInProgress Card)))
                                      (NEQ OpInProgress 'US))
                                  (NC.PrintOperationInProgressMsg InstigatingWindow "Select Card" OpInProgress)
                                  (DISMISS 1000)
                                  (WINDOWPROP PromptWindow 'SelectedCards (SETQ SelectedCards (CDR SelectedCards)))
                                  (NC.ClearMsg InstigatingWindow NIL))
                                  [OR (NOT SingleCardFlg)
                                    FirstTime)
                                  ; A valid selection.
                                  (SETQ SelectedCard Card)
                                  (SETQ FirstTime NIL)
                                  (NC.ClearMsg InstigatingWindow NIL)
                                  [if FileLevelLockFlg
                                    then [SETQ ResetItems (LIST (NAMED-RESETSAVE
                                      CardListResetVar
                                      (NC.NoteFileProp (fetch (Card NoteFile)
                                        of Card)
                                          'OperationInProgress "Select Card")
                                          '(NC.NoteFileProp , (fetch (Card NoteFile)
                                            of Card)
                                              OperationInProgress NIL))
                                      (NAMED-RESETSAVE
                                        CardListResetVar
                                        (NC.NoteFileProp (fetch (Card NoteFile)
                                          of Card)
                                            of Card)

```





```

    else (DREVERSE SelectedCards]
    (WINDOWPROP PromptWindow 'SelectedCards NIL))])

```

**(NC.BinLoopProcess**

[LAMBDA NIL

; Edited 3-Mar-89 16:33 by krivacic

```

;;; bk: 2/7/89 Just Wait for a CR keydown & TTY.PROCESSP. This avoids problems when shift-selecting thinks with a CR in them.

```

```

;;; bk: 3/3/89 Replace the BIN with \GETKEY. Many Thanks to BvM if this works.

```

```

    (PROG ((THIS-WINDOW (PROCESSPROP (THIS.PROCESS)
                                     'WINDOW))
          VAL)
          (BLOCK)
          (TTYDISPLAYSTREAM THIS-WINDOW)
          ;; Wait for a Carriage Return
          (while [OR (NOT (TTY.PROCESSP))
                    (WINDOWPROP THIS-WINDOW 'ShiftSelectingCards)
                    (NOT (KEYDOWNP 'CR])
                do (\GETKEY)
                  (BLOCK))
          ;; Indicate to the Menu Process that a CR was received
          [WINDOWPROP THIS-WINDOW 'SelectedCards (CONS 'DONE (WINDOWPROP THIS-WINDOW 'SelectedCards)
              (NOTIFY.EVENT (WINDOWPROP THIS-WINDOW 'CopyInsertEvent))
          ;; Block till the Menu Process kills this process
          (while T do (BLOCK))

```

**(NC.ZapBinLoopProcess**

[LAMBDA (Window)

(\* Randy.Gobbel "12-Aug-87 16:15")

```

    (LET [(BinLoopProcess (WINDOWPROP Window 'PROCESS]
          (WINDOWDELPROP Window 'CLOSEFN (FUNCTION NC.ZapBinLoopProcess))
          (AND (PROCESSP BinLoopProcess)
              (DEL.PROCESS BinLoopProcess]))

```

**(NC.SelectNoteCardsCopyInsertFn**

[LAMBDA (ImageObj Window)

(\* rht%: "8-Jun-87 11:15")

```

    (* Copy insert fn for promptwindow during SelectNoteCards operation)

```

```

    (LET [(CardProcessedEvent (WINDOWPROP Window 'CardProcessedEvent))
          (SelectCardsMonitor (WINDOWPROP Window 'SelectCardsMonitor]
          (WITH.MONITOR SelectCardsMonitor
            (until (OR [NOT (PROCESSP (WINDOWPROP Window 'SelectNoteCardsProcess]
                          (WINDOWPROP Window 'NewCardsProcessed))
                      do (MONITOR.AWAIT.EVENT SelectCardsMonitor CardProcessedEvent 1000))
            (LET [[Card (COND
                    ((NC.LinkIconImageObjP ImageObj)
                     (fetch (Link DestinationCard) of (NC.FetchLinkFromLinkIcon ImageObj)))
                    ((NC.CardImageObjP ImageObj)
                     (IMAGEOBJPROP ImageObj 'OBJECTDATUM]
                    (SelectedCards (WINDOWPROP Window 'SelectedCards))
                    (CopyInsertEvent (WINDOWPROP Window 'CopyInsertEvent]
                    (* * Add the selected acrd to the selected cards list)
                    (if (AND Card (for SelectedCard in SelectedCards never (NC.SameCardP Card SelectedCard)))
                        then (WINDOWPROP Window 'SelectedCards (CONS Card SelectedCards))
                          (WINDOWPROP Window 'NewCardsProcessed NIL))
                    (* * Wake up the main process)
                    (NOTIFY.EVENT CopyInsertEvent T)))]))

```

**(NC.CopyButtonEventFn**

[LAMBDA (Window)

; Edited 10-Jan-89 14:50 by rtk

```

;;; Return NoteCardsObject if button is in title bar, else call the old copy button event fn or just the button event fn.

```

```

;;; fgh 1/30/86 Added check to make sure that the OldCopyButtonEventFn is not this function before it is APPLIED. Otherwise you get an infinite
;;; recursion.

```

```

;;; bk 1/10/89 Add ShiftSelectingCards property so we can stay in NC.SelectNoteCards while shift selection is in progress.

```

```

    (LET (CopyButtonEventFn TTYWindow)
      (if (NOT (INSIDEP (DSPCLIPPINGREGION NIL (WINDOWPROP Window 'DSP))
                        (LASTMOUSEX Window)
                        (LASTMOUSEY Window)))
          then (until (OR (MOUSESTATE UP)
                          (NEQ Window (WHICHW))
                          (INSIDEP (DSPCLIPPINGREGION NIL (WINDOWPROP Window 'DSP))

```

```

                (LASTMOUSEX Window)
                (LASTMOUSEY Window))
    do (BLOCK))
[if (AND (NOT (INSIDEP (DSPCLIPPINGREGION NIL (WINDOWPROP Window 'DSP))
                    (LASTMOUSEX Window)
                    (LASTMOUSEY Window)))
      (MOUSESTATE UP))
  then (if (AND [WINDOWP (SETQ TTYWindow (PROCESSPROP (TTY.PROCESS)
                                                    'WINDOW]
                    (WINDOWPROP TTYWindow 'SelectingCards))
            then (COPYINSERT (NC.MakeCardImageObject (WINDOWPROP Window 'NoteCardObject]
else (SETQ TTYWindow (PROCESSPROP (TTY.PROCESS)
                                'WINDOW))
  (AND TTYWindow (WINDOWPROP TTYWindow 'ShiftSelectingCards T))
  (if (AND (SETQ CopyButtonEventFn (WINDOWPROP Window 'OldCopyButtonEventFn))
        (NEQ CopyButtonEventFn 'NC.CopyButtonEventFn))
    then (APPLY* CopyButtonEventFn Window)
  elseif (SETQ CopyButtonEventFn (WINDOWPROP Window 'BUTTONEVENTFN))
    then (APPLY* CopyButtonEventFn Window)
  (AND TTYWindow (WINDOWPROP TTYWindow 'ShiftSelectingCards NIL])

```

**(NC.ShrukenIconButtonEventFn**

; Edited 9-Jan-89 12:46 by rtk

```

[LAMBDA (ICON-WINDOW)
  (LET (CopyButtonEventFn TTYWINDOW)
    ;; In Shift Select
    (if (AND (OR (KEYDOWNP 'RSHIFT)
                  (KEYDOWNP 'LSHIFT))
            (OR (MOUSESTATE LEFT)
                  (MOUSESTATE MIDDLE)))
      then
        ;; Wait for the Mouse Up
        (until (OR (MOUSESTATE UP)
                    [NOT (OR (KEYDOWNP 'RSHIFT)
                              (KEYDOWNP 'LSHIFT])
                    (NEQ ICON-WINDOW (WHICHW)))
              do (BLOCK))
        ;; If Mouse UP & Shift Select
        (if (MOUSESTATE UP)
          then [LET* ((SELECT-WINDOW (PROCESSPROP (TTY.PROCESS)
                                                    'WINDOW))
                     (PARENT-WINDOW (WINDOWPROP ICON-WINDOW 'ICONFOR))
                     (THIS-CARD (NC.CoerceToCard PARENT-WINDOW))
                     (DEST-CARD LINK-OBJ LINK))
            (if (AND (WINDOWP SELECT-WINDOW)
                    (WINDOWPROP SELECT-WINDOW 'SelectingCards))
              then [COPYINSERT (NC.MakeCardImageObject (WINDOWPROP PARENT-WINDOW
                                                                    'NoteCardObject]
              else (if (AND THIS-CARD (NC.CoerceToCard SELECT-WINDOW))
                      then (SETQ LINK (NC.MakeLink SELECT-WINDOW 'Unspecified THIS-CARD
                                                    (NC.CoerceToCard SELECT-WINDOW)))
                      (COPYINSERT (NC.MakeLinkIcon LINK))
                      else (PROMPTPRINT "Destination of Link not a NoteCard"]
            else (ICONBUTTONEVENTFN ICON-WINDOW))
          else (ICONBUTTONEVENTFN ICON-WINDOW])

```

**(NC.CardImageObjP**

(\* fgh%: "19-Dec-85 23:34")

```

[LAMBDA (ImageObj)
  (type? Card (IMAGEOBJPROP ImageObj 'OBJECTDATUM])

```

**(NC.MakeCardImageObject**

(\* Randy.Gobbel "5-Nov-86 15:49")

```

[LAMBDA (CardObject)
  (DECLARE (GLOBALVARS NC.CardImageObjFns))
  (IMAGEOBJCREATE CardObject NC.CardImageObjFns])

```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.CardImageObjFns)

)

;;; Fix up Lisp

```

[LOADINITADVISE NC-ADVICE0260 (LOGOUT (NIL (BEFORE NIL (if (NULL (NC.LogoutAdvice))
                    then (RETURN]

```

;;; Stuff for creating and operating the NoteCards session icon.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```
{MEDLEY}<notecards>system>NCINTERFACE.;1
```

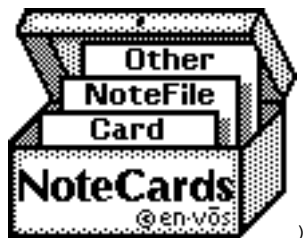
Page 28

```
(GLOBALVARS NC.CardOpsMenu NC.CardOpsItems NC.NoteFileOpsMenu NC.NoteFileOpsItems NC.OtherOpsMenu
  NC.OtherOpsItems NC.NoteCardsIconBitmap)
)
```

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

```
;;; Bitmaps
```

```
(RPAQQ NC.NoteCardsIconBitmap
```



```
)
```

```
(RPAQQ NC.NewCardsIconBitmap
```



```
)
```

```
(RPAQQ NC.BoxesIconBitmap
```



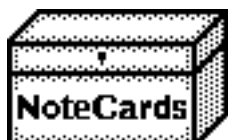
```
)
```

```
(RPAQQ NC.NoteCardsIconMask
```



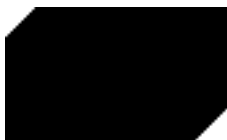
```
)
```

```
(RPAQQ NC.NoteCardsIconShrunkenBitmap
```



```
)
```

```
(RPAQQ NC.NoteCardsIconShrunkenMask
```



```
)
```

```
;;;
```

```
(DEFINEQ
```

```
(NC.NoteCardsWindowP
```

```
  [LAMBDA (Window)
```

; Edited 8-Dec-88 11:28 by krivacic

```
;;; Is this some sort of NoteCards window?
```

```
  (OR (NC.CoerceToCard Window)
       (NC.NoteFileMenuWindowP Window)
       (NC.NoteCardsIconWindowP Window))
```

```
(NC.NoteCardsIconWindowP
```

```
  [LAMBDA (Window)
```

(\* fgh%: "25-Jun-86 19:43")

```
    (* * Is this the NC session icon window?)
```

```
    (* * fgh |6/25/86| First created.)
```

(EQ Window NC.NoteCardsIconWindow))

**(NC.MakeNoteCardsIcon**

[LAMBDA (Position Style)

(\* pmi%: "18-Mar-87 17:15")

(\* \* This creates and returns a window containing the NoteCards icon bitmap.)

(\* \* pmi 3/11/87%: Revamped for another new session icon.)

```
(SELECTQ Style
  (OldStyle (NC.MakeOldStyleNoteCardsIcon Position))
  (NewStyle (NC.MakeNewStyleNoteCardsIcon Position))
  (NC.MakeNewStyleNoteCardsIcon Position))
```

**(NC.BringUpNoteCardsIcon**

[LAMBDA (Position Style)

(\* pmi%: " 8-Dec-87 15:24")

(\* \* Either flash existing NoteCards icon or make a new one.)

(\* \* fgh |6/7/86| Added Position arg to pass to NC.MakeWindow)

(\* \* kirk 1Jul86 Changed FLASHW to FLASHWINDOW)

(\* \* pmi 3/18/87%: Added Style argument to be passed to NC.MakeNoteCardsIcon for selection of session icon style)

(\* \* pmi 12/8/87%: Added check for NC.NoteCardsIconWindow being bound.  
Also, now returns NC.NoteCardsIconWindow.)

```
(DECLARE (GLOBALVARS NC.NoteCardsIconWindow))
(if (AND (BOUNDP 'NC.NoteCardsIconWindow)
  (WINDOWP NC.NoteCardsIconWindow))
  then (if Position
    then (MOVEW NC.NoteCardsIconWindow Position)
    (FLASHWINDOW NC.NoteCardsIconWindow))
  else (SETQ NC.NoteCardsIconWindow (NC.MakeNoteCardsIcon Position Style)))
NC.NoteCardsIconWindow))
```

**(NC.NoteCardsIconAfterMoveFn**

[LAMBDA (Window)

(\* pmi%: "26-Mar-87 17:14")

(\* \* Updates NC.NoteCardsIconPosition when the session icon is moved.)

```
(DECLARE (GLOBALVARS NC.NoteCardsIconPosition))
(SETQ NC.NoteCardsIconPosition (CREATEPOSITION (fetch (REGION LEFT) of (WINDOWPROP Window 'REGION))
  (fetch (REGION BOTTOM) of (WINDOWPROP Window 'REGION))
```

**(NC.FileBrowserMenu**

[LAMBDA (Window InSelection InRegion)

; Edited 8-Oct-2021 13:49 by nc

;; Bring up a notecards file browser after user selects pattern.

;; rht 7/2/86: Now calls NC.GetFileBrowserPattern. Also only stores file browser wins on the session icon windowprop rather than both wins and  
;; patterns.

;; kef 8/8/86: Added kludge for constructing file browsers on remote Server.

;; pmi 3/25/87: Added NC.MenuFont and cleaned up menu for consistency.

;; fgh 2021-10-08: Added InSelection and InPosition arguments so that we can call from init file on Notecards Online

```
(DECLARE (GLOBALVARS NC.MenuFont NC.FileBrowserPatterns NC.FileBrowserDefaultPatterns
  NC.FileBrowserMenuItems))
(LET (Menu Selection)
  [if InSelection
    then (SETQ Selection InSelection)
    else (PROGN (SETQ Menu (create MENU
      ITEMS _ [APPEND (UNION NC.FileBrowserPatterns
        NC.FileBrowserDefaultPatterns)
        ' (|--- New Pattern ---| 'New% Pattern "Make a new
          Notefile browser pattern."])
      TITLE _ " Notefile Browser Pattern "
      MENUFONT _ NC.MenuFont
      CENTERFLG _ T
      ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
        1)))
    (if (EQ (SETQ Selection (MENU Menu))
      'New% Pattern)
      then ; User wants to give us a new pattern.
        (SETQ Selection (NC.GetFileBrowserPattern Window NIL])
    (if Selection
      then ;; See if browser for that pattern already exists, else create one and stash on the NC icon's window's proplist.
        (OR (NC.FlashFileBrowserWin Selection Window)
```

(NC.ListOfNoteFilesMenu

```

(DECLARE (GLOBALVARS NC.MenuFont NCP.NoticedNoteFileNames))
(LET (Menu Items Result)
  [SETQ Items `(@ (for NoteFileName in NCP.NoticedNoteFileNames bind NoteFile Stream RootName
    when (PROGN (SETQ NoteFile (NC.NoteFileFromFileName NoteFileName))
      (OR (AND (EQ ShowOnlyOpenNFsFlg 'CLOSED)
        (NULL (NC.NoteFileOpenP NoteFile)))
        (NULL ShowOnlyOpenNFsFlg)
        (NC.NoteFileOpenP NoteFile)))
    collect (LIST [CONCAT (if (NC.NoteFileOpenP NoteFile)
      then
        (* (if (WINDOWP (WFROMMENU
          (fetch (NoteFile Menu) of NoteFile))) then "*" else "$"))
        " ")
      else " ")
      (SUBSTRING [SETQ RootName (PACKFILENAME 'NAME (FILENAMEFIELD
        NoteFileName
        'NAME)
        'VERSION
        (FILENAMEFIELD NoteFileName
        'VERSION)]
      1
      (MIN 20 (NCHARS RootName]
      NoteFileName
      (CONCAT "Selects NoteFile " NoteFileName)))
    ,@ (if IncludeNewNoteFileFlg
      then (LIST '("-- Other NoteFile --" 'NEW "Select some other notefile - you'll be
        prompted for the name."))
      else NIL]
  [SETQ Result (if (NULL Items)
    then NIL
    elseif [AND (EQ (LENGTH Items)
      1)
      (EQUAL (CADAR Items)
      'NEW]
    then 'NEW
    else (MENU (create MENU
      ITEMS _ Items
      TITLE _ (OR Operation "NoteFiles")
      MENUFONT _ NC.MenuFont
      ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
      1]
    (if (EQ Result 'NEW)
      then (SETQ Result (NC.DatabaseFileName (CONCAT "Name of NoteFile to " (SUBSTRING Operation 1 -9)
        (CHARACTER 13)))
        " -- " T T NIL InterestedWindow)))

```

Result))

**(NC.CheckpointSession**

[LAMBDA NIL

(\* kirk%: "20-Jan-86 00:53")

(\* \* Checkpoint all open NoteFiles)

```
[MAPHASH NC.NoteFilesHashArray (FUNCTION (LAMBDA (Value Key)
  (AND (type? NoteFile Value)
    (if (AND (STREAMP (SETQ Stream (fetch (NoteFile Stream)
      of Value)))
      (OPENP Stream))
    then (NC.CheckpointDatabase Value]
(NC.PrintMsg NIL T "Done checkpointing entire session."))]
```

**(NC.CloseSession**

[LAMBDA (DeleteChangesFlg AutoConfirmFlg)

(\* Randy.Gobbel "4-Mar-87 16:18")

(\* \* Checkpoint all open NoteFiles)

(\* \* rht 2/16/87%: Now accepts AutoConfirmFlg and passes to NC.AbortSession and NC.CloseNoteFile.)

```
[MAPHASH NC.NoteFilesHashArray (FUNCTION (LAMBDA (Value Key)
  (AND (type? NoteFile Value)
    (if (AND (STREAMP (SETQ Stream (fetch (NoteFile Stream)
      of Value)))
      (OPENP Stream))
    then (if DeleteChangesFlg
      then (NC.AbortSession Value NIL AutoConfirmFlg)
      else (NC.CloseNoteFile Value NIL AutoConfirmFlg]
(NC.PrintMsg NIL T "Done closing all NoteFiles."))]
```

**(ADDTOVAR NC.NoteFileIconOperationsMenuItems**

```
(Open NIL "Opens this NoteFile." (SUBITEMS (Open% Read/Write NIL "Opens a notefile for Read/Write
  access."
  (Open% Read-only NIL "Open a notefile for read-only.")))
(Checkpoint NIL "Checkpoint this NoteFile, saving changed cards.")
(Close NIL "Closes this NoteFile." (SUBITEMS (Close% NoteFile NIL "Close a notefile."
  (|Close w/o confirm| NIL "Close this notefile w/o asking for
    user confirm before closing open cards.")))
(Abort NIL "Close NF, deleting changes since last checkpoint.")
(Compact NIL "Compacts this NoteFile to a new file." (SUBITEMS (|Compact To New File| NIL "Compact a
  NoteFile copying to a new
  file."
  (|Compact In Place| NIL "Compact a
    NoteFile w/o copying to a new
    file.")))
(Inspect&Repair NIL "Inspects and optionally repairs this NoteFile." (SUBITEMS (From% Links NIL "Do an
  Inspect & Repair
  on a NoteFile
  from the Link
  information."
  (From% Contents NIL
    "Inspects and
    optionally repairs
    a Notefile, but
    reads every card
    substance. This
    slows it WAY
    down.")))
(Copy NIL "Copies this notefile to a target file.")
(Rename NIL "Rename this NoteFile")
>Delete NIL "Deletes this NoteFile."))
```

(ADDTOVAR **NC.NoteFileIconOpenOperations** Checkpoint Close Close% Notefile |Close w/o confirm| Abort)

```
(ADDTOVAR NC.NoteFileIconCloseOperations Open Open% Read/Write Open% Read-only Compact |Compact To New File|
  |Compact In Place| Inspect&Repair From% Links From% Contents Copy
  Rename Delete)
```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```
(GLOBALVARS NC.InitialNoteFileIconOperationsMenuItems NC.InitialNoteFileIconOpenOperations
  NC.InitialNoteFileIconCloseOperations NC.ShowPropListMenu NC.EditPropListMenu)
)
```

(RPAQ? **NC.InitialNoteFileIconOperationsMenuItems** NC.NoteFileIconOperationsMenuItems)(RPAQ? **NC.InitialNoteFileIconOpenOperations** NC.NoteFileIconOpenOperations)(RPAQ? **NC.InitialNoteFileIconCloseOperations** NC.NoteFileIconCloseOperations)(RPAQ? **NC.DefaultMiddleButtonMenuItems** NIL)

(RPAQ? **NC.ShowPropListMenu**

```
(create MENU ITEMS _ ' ( ("Quit" (NC.ClosePropListEditor W 'NoSave)
                                "Quit from pointer list display.))
                                CENTERFLG _ T MENUFONT _ NC.MenuFont ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                1)))
```

(RPAQ? **NC.EditPropListMenu**

```
(create MENU ITEMS _ ' ( ("Add New Property" (NC.AddPropToPropList W)
                                "Add a new property to this card's property list.")
                                ("Delete Selected Property" (NC.DelPropFromList W)
                                "Delete selected property from this card's property list.")
                                ("Quit w/o Saving Changes" (NC.ClosePropListEditor W 'NoSave)
                                "Quit from property list edit. Changes are not saved.")
                                ("Quit - Saving Changes" (NC.ClosePropListEditor W 'Save)
                                "Quit from property list editor. Save changes.))
                                CENTERFLG _ T MENUFONT _ NC.MenuFont ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                1)))
```

(RPAQ? **NC.LocalInsertLinkMenuItem**

```
' (Insert% Link (FUNCTION NC.AddLinkToCard)
    "Insert a link to another card at the currently selected point in the text."
    (SUBITEMS (Insert% Link (FUNCTION NC.AddLinkToCard)
        "Insert a link to another card at the currently selected point in the text."
        (Insert% Links (FUNCTION NC.AddLinksToCard)
            "Insert links to other cards at the currently selected point in the text."
            (|Add Global Link| (FUNCTION NC.AddGlobalLinkToCard)
                "Add a global link emanating from this card.")
            (|Add Global Links| (FUNCTION NC.AddGlobalLinksToCard)
                "Add multiple global links emanating from this card.))))
```

(RPAQ? **NC.GlobalInsertLinkMenuItem**

```
' (|Add Global Link| (FUNCTION NC.AddGlobalLinkToCard)
    "Add a global link emanating from this card."
    (SUBITEMS (|Add Global Link| (FUNCTION NC.AddGlobalLinkToCard)
        "Add a global link emanating from this card."
        (|Add Global Links| (FUNCTION NC.AddGlobalLinksToCard)
            "Add multiple global links emanating from this card.))))
```

(RPAQ? **NC.DefaultLeftButtonMenuItems**

```
` ( (|Edit Property List| (FUNCTION NC.EditProperties)
    "Brings up an editor for the property list of this card."
    (Show% Links (FUNCTION NC.ShowLinks)
        "Brings up a list of the links to and from this card."
    (Show% Info (FUNCTION NC.ShowInfo)
        "Brings up a window containing special information about this card."
        (SUBITEMS (Indicate% NoteFile (FUNCTION NC.AttachNoteFileName)
            "Shows the name of this card's NoteFile.))
    (Designate% FileBoxes (FUNCTION NC.AddParents)
        "File this note card in one or more file boxes."
        (SUBITEMS (|Unfile from FileBoxes| (FUNCTION NC.UnfileNoteCard)
            "Remove this card from one or more of its file boxes.))
    (Assign% Title (FUNCTION NC.AssignTitle)
        "Assigns a (new) title to this note card."
    (Title/FileBoxes [FUNCTION (LAMBDA (TextStream)
        (NC.AssignTitle TextStream)
        (NC.AddParents TextStream)
        "Do both assigning of title and filing in fileboxes."
    ,NC.LocalInsertLinkMenuItem
    (|Close and Save| (FUNCTION CLOSEW)
        "Close this note card after saving it in the NoteFile."
        (SUBITEMS (|Close and Save| (FUNCTION CLOSEW)
            "Close this note card after saving it in the NoteFile."
            (|Close w/o Saving| (FUNCTION NC.QuitWithoutSaving)
                "Close this note card without saving any changes made since the last Save."
            (|Save w/o Closing| (FUNCTION NC.CardSaveFn)
                "Save this card in the NoteFile but don't close the card."
            (Delete% Card (FUNCTION NC.DeleteNoteCard)
                "Permanently delete this card from the NoteFile.])
```

(ADDTTOVAR **NC.CardOpsItems**

```
(| Close | (NC.CloseNoteCards NIL NC.NoteCardsIconWindow)
    "Close several note cards."
    (SUBITEMS (Close% NoteCards (NC.CloseNoteCards NIL NC.NoteCardsIconWindow)
        "Close several note cards."
        (Close% Structure (NC.CloseStructure NIL NIL NC.NoteCardsIconWindow)
            "Close note cards belonging to a structure.))
    (| Delete | (NC.DeleteNoteCards NIL T NIL NC.NoteCardsIconWindow)
        "Delete several note cards."
        (SUBITEMS (Delete% NoteCards (NC.DeleteNoteCards NIL T NIL NC.NoteCardsIconWindow)
            "Delete several note cards."
            (DeleteStructure (NC.DeleteStructure NIL NIL NC.NoteCardsIconWindow)
                "Delete note cards belonging to a structure.))
    (| Copy | (NC.CopyCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
        "Copy note cards to a filebox"
        (SUBITEMS (Copy% Cards (NC.CopyCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
            "Copy note cards to a filebox"
            (Copy% Structure (NC.CopyStructure NIL NIL NIL NC.NoteCardsIconWindow)
```



```

        "Copy note cards belonging to a structure.")))
(| Move | (NC.MoveCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
  "Move note cards to a filebox"
  (SUBITEMS (Move% Cards (NC.MoveCards NIL NIL NIL NIL NC.NoteCardsIconWindow)
    "Move note cards to a filebox")
    (Move% Structure (NC.MoveStructure NIL NIL NIL NC.NoteCardsIconWindow)
      "Move note cards belonging to a structure.")))

(ADDTOVAR NC.NoteFileOpsItems
(| Open | (NC.DoNoteFileOp 'Open% NoteFile)
  "Opens a notefile for Read/Write access."
  (SUBITEMS (Open% Read/Write (NC.DoNoteFileOp 'Open% NoteFile)
    "Opens a notefile for Read/Write access."
    (Open% Read-only (NC.DoNoteFileOp 'Read-only% Open)
      "Open a notefile for read-only.")))
(| Compact | (NC.DoNoteFileOp 'Compact% NoteFile)
  "Compact a NoteFile copying to a new file."
  (SUBITEMS (|Compact To New File| (NC.DoNoteFileOp 'Compact% NoteFile)
    "Compact a NoteFile copying to a new file."
    (|Compact In Place| (NC.DoNoteFileOp 'Compact In Place|)
      "Compact a NoteFile w/o copying to a new file.")))
(| Inspect&Repair | (NC.DoNoteFileOp 'Inspect&Repair% NoteFile)
  "Do an Inspect & Repair on a NoteFile from the Link information."
  (SUBITEMS (From% Links (NC.DoNoteFileOp 'NC.DoNoteFileOp)
    "Do an Inspect & Repair on a NoteFile from the Link information."
    (From% Contents (NC.DoNoteFileOp 'Read% Substances)
      "Inspects and optionally repairs a Notefile, but reads every substance. This
      slows it WAY down.")))
(| Copy | (NC.DoNoteFileOp 'Copy% NoteFile)
  "Copy a NoteFile."
(| Rename | (NC.DoNoteFileOp 'Rename% NoteFile)
  "Rename a NoteFile")
(| Delete | (NC.DoNoteFileOp 'Delete% NoteFile)
  "Delete a NoteFile")
(| Create | (NC.DoNoteFileOp 'Create% NoteFile)
  "Create a new NoteFile w/o opening it."
  (SUBITEMS (| Create without Open | (NC.DoNoteFileOp 'Create% NoteFile)
    "Create a new NoteFile w/o opening it."
    (| Create and Open | (NC.DoNoteFileOp 'Create and Open|)
      "Create a new NoteFile and open it.")))
(| ----- | NIL "")
(| Checkpoint | (NC.DoNoteFileOp 'Checkpoint% NoteFile)
  "Checkpoint a NoteFile"
  (SUBITEMS (Checkpoint% Notefile (NC.DoNoteFileOp 'Checkpoint% NoteFile)
    "Checkpoint a NoteFile."
    (Checkpoint% All (NC.CheckpointSession)
      "Checkpoint all currently open NoteFiles"))
(| Close | (NC.DoNoteFileOp 'Close% NoteFile)
  "Close a notefile."
  (SUBITEMS (Close% Notefile (NC.DoNoteFileOp 'Close% NoteFile)
    "Close a notefile."
    (|Close w/o confirm| (NC.DoNoteFileOp 'Close w/o confirm|)
      "Close a notefile w/o asking for user confirm before closing open cards.")
    (Close% All (NC.CloseSession)
      "Close all currently open NoteFiles")
    (|Close All w/o confirm| (NC.CloseSession NIL T)
      "Close all open notefiles w/o asking for user confirm before closing open
      cards.")))
(| Abort | (NC.DoNoteFileOp 'Abort% NoteFile)
  "Close NF, deleting changes since last checkpoint."
  (SUBITEMS (Abort% Notefile (NC.DoNoteFileOp 'Abort% NoteFile)
    "Close Notefile, delete changes since last checkpoint."
    (Abort% All (NC.CloseSession T)
      "Abort all currently open NoteFiles"))
(| ----- | NIL "")
(| NC FileBrowser | (NC.FileBrowserMenu NC.NoteCardsIconWindow)
  "Bring up a NoteFile browser."))

```

## (ADDTOVAR NC.OtherOpsItems

```

(| Edit Parameters | (NC.BuildInspector)
  "Adjust NoteCards profile for this session.")
(| NF Indicators On | (NC.ShowNoteFiles)
  "Turn on the NoteFile indicator for all currently open cards"
  (SUBITEMS (Indicators% On (NC.ShowNoteFiles)
    "Turn on the NoteFile indicator for all currently open cards"
    (Indicators% Off (NC.ShowNoteFiles T)
      "Turn off any NoteFile indicator on each card on the screen.")))
(| TEdit Killer On | (START-TEDIT-KILLER)
  "Start the TEDIT-KILLER process."
  (SUBITEMS (| TEdit Killer On | (START-TEDIT-KILLER)
    "Start the TEDIT-KILLER process."
    (|TEdit Killer Off| (STOP-TEDIT-KILLER)
      "Stop the TEDIT-KILLER process."
    (|New Process Limit| (SETQ TEDIT-PROCESS-LIMIT (OR (NUMBERP (RNUMBER (CONCAT "Old Tedit
process
limit: "
TEDIT-PROCESS-LIMIT

```

```

)
NIL NIL NIL T))
TEDIT-PROCESS-LIMIT))

"Change value of TEdit process limit."
(|New Wait Time| (SETQ TEDIT-KILLER-WAIT-TIME
                    (OR (LET* [(MULTIPLIER 100)
                                (NEW-WAIT-TIME (NUMBERP (RNUMBER (CONCAT "Current
                                                                    TEdit killer
                                                                    wait time: "
                                                                    (IQUOTIENT
                                                                    TEDIT-KILLER-WAIT-TIME
                                                                    100)
                                                                    " sec"))
                                                                    NIL NIL NIL T]
                            (AND NEW-WAIT-TIME (ITIMES NEW-WAIT-TIME MULTIPLIER)))
                        TEDIT-KILLER-WAIT-TIME)))
                    "Change value of TEdit killer wait time.")))
(| Load Patches | (NC.LoadPatches)
                    "Load the latest NoteCards patches."))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.InitialCardOpsItems NC.InitialNoteFileOpsItems NC.InitialOtherOpsItems)
)

(RPAQ? NC.InitialCardOpsItems NC.CardOpsItems)

(RPAQ? NC.InitialNoteFileOpsItems NC.NoteFileOpsItems)

(RPAQ? NC.InitialOtherOpsItems NC.OtherOpsItems)

(RPAQ? NC.CardOpsMenu )

(RPAQ? NC.OtherOpsMenu )

(RPAQ? NC.NoteFileOpsMenu )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.NoteCardsIconMenuBitMap NC.OldStyleIconMenuBitmap NC.OldStyleIconBitmap NC.OldStyleIconMenuBitmap
NC.NoteCardsIconMask NC.NoteCardsIconShrunkenBitmap NC.NoteCardsIconShrunkenMask NC.OtherOpsItems
NC.OtherOpsMenu)
)

(RPAQ? NC.NoteCardsIconMenuBitMap )

(RPAQ? NC.OldStyleIconMenuBitmap )

(DEFINEQ

(NC.RedisplayNoteCardsIconWindow
[LAMBDA (Window)
(* pmi%: "18-Mar-87 17:20")

(* * Redisplay the NC Icon window. Use different displkays whether the cursor is on or out of the window)

(* * fgh |6/23/86| First created.)

(* * pmi 3/18/87%: Now used only for the old style session icon.
Bitmap names needed to change.)

(DECLARE (GLOBALVARS NC.OldStyleIconMenuBitmap NC.OldStyleIconBitmap))
(if (NULL (WINDOWPROP Window 'NoteCardsOperationInProgress))
    then (GETMOUSESTATE)
        (if (INSIDEP (WINDOWPROP Window 'REGION)
                LASTMOUSEX LASTMOUSEY)
            then (BITBLT NC.OldStyleIconMenuBitmap 0 0 Window 0 0 NIL NIL 'INPUT 'REPLACE)
            else (BITBLT NC.OldStyleIconBitmap 0 0 Window 0 0 NIL NIL 'INPUT 'REPLACE]))

(NC.NoteCardsIconButtonEventFn
[LAMBDA (Window)
; Edited 3-Dec-87 18:59 by rht:

(* * Bring up a menu of all notefiles found in the notefiles hash array.
Also allow user to open a new notefile.)

(* * rht 1/7/86%: Now handles middle button differently -
brings up menu of patterns and creates notefile browser for that pattern.)

(* * kirk 18Jan85 Added session commands to left button. Carved out NC.CommandMenu and NC.FileBrowserMenu
functions.)

(* * fgh |6/24/86| Totally revamped for new 1.3 session icon. Now is a whenselectedfn for the icon.
Pops up the menu for the selectedfn, creating one if necessary.)

(* * pmi 3/11/87%: Revamped for another new session icon.)

```

(\* \* pmi 3/25/87%: Added NC.MenuFont to all menus)

```
(DECLARE (GLOBALVARS NC.MenuFont NC.NoteCardsIconWindow))
(if (MOUSESTATE UP)
  else (TOTOPW Window)
        (ALLOW.BUTTON.EVENTS)
        (LET [(Menus (WINDOWPROP Window 'MENU])
```

(\* \* For the new 1.3 icon, there are three menus in the icon window.  
Each is a menu of one item.)

```
(for Menu in Menus
  when (for Item in (fetch (MENU ITEMS) of Menu) when (INSIDEP (MENUITEMREGION Item Menu)
                                                                (LASTMOUSEX Window)
                                                                (LASTMOUSEY Window)))
    do (SHADEITEM Item Menu BLACKSHADE) (* RESETSAVE (SHADEITEM Item Menu BLACKSHADE)
                                                    (BQUOTE (SHADEITEM (\, Item)
                                                                (\, Menu) (\, WHITESHADE))))

      [MENU (OR (EVAL (CADR Item))
                (SET (CADR Item)
                    (create MENU
                        ITEMS _ (EVAL (CADDR Item))
                        MENUFONT _ NC.MenuFont
                        CHANGEOFFSETFLG _ 'Y
                        MENUOFFSET _ (CONS -1 0)
                        TITLE _ (CONCAT " " (CAR Item)
                                         " Ops ")
                        CENTERFLG _ T
                        ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
                                             1])
                    (REDISPLAYW NC.NoteCardsIconWindow)
                    (RETURN T))
                do (RETURN)])
```

## (NC.DoNoteFileOp

[LAMBDA (Op)

; Edited 23-Feb-89 11:04 by krivacic

;; Do a NoteFile op chosen from NC icon menu

;; rht 7/2/86: Now calls NC.AbortSession with NC.NoteCardsIconWindow arg.

;; rht 7/5/86: Added Read-only Open entry.

;; rht 7/17/86: Now calls NC.InspectAndRepairNoteFile instead of NC.ScavengerPhase1.

;; pmi 12/4/86: Added NC.NoteCardsIconWindow argument to calls to NC.ListOfNoteFilesMenu for 'Delete' and 'Rename' operations

;; rht 2/11/87: Now handles case when Op has unexpected value.

;; rht 2/16/87: Added Close w/o confirm case.

;; pmi 5/15/87: overhauled to use NC.NoticedNoteFileNamesMenu instead of NC.ListOfNoteFilesMenu. Added 'Create and Open' and 'Create and Read-only Open' options.

;; pmi 9/3/87: Now passes NC.NoteCardsIconWindow for InterestedWindow in calls to Compact functions.

;; bk 2/23/89: Add NC.BringUpTOC parm to NC.OpenDatabaseFile

```
(DECLARE (GLOBALVARS NC.NoteCardsIconWindow NC.MsgDelay NC.BringUpTOC))
(SELECTQ Op
  (Open% NoteFile
    (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow 'Open% NoteFile]
            (if (NULL NoteFileName)
                then NIL
                else (if (EQ NoteFileName 'NEW)
                        then (SETQ NoteFileName NIL)
                        (NC.OpenDatabaseFile NoteFileName NIL NIL NIL NIL NIL NIL NIL NIL
                                              NC.NoteCardsIconWindow NC.BringUpTOC))))
    (Read-only% Open
      (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow 'Open% NoteFile]
            (if (NULL NoteFileName)
                then NIL
                else (if (EQ NoteFileName 'NEW)
                        then (SETQ NoteFileName NIL)
                        (NC.OpenDatabaseFile NoteFileName 'INPUT NIL NIL NIL NIL NIL NIL NIL NIL
                                              NC.NoteCardsIconWindow NC.BringUpTOC))))
    (Checkpoint% NoteFile
      (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu NIL 'OPEN NC.NoteCardsIconWindow
                          'Checkpoint% NoteFile]
            (if NoteFileName
                then (NC.CheckpointDatabase (NC.NoteFileFromFileName NoteFileName)
                                             NIL NIL NC.NoteCardsIconWindow)
                else (NC.PrintMsg NC.NoteCardsIconWindow NIL "Checkpoint cancelled." (CHARACTER 13))
                     (DISMISS NC.MsgDelay)
                     (NC.ClearMsg NC.NoteCardsIconWindow T))))
    (Close% NoteFile
```

```

(LET [(NoteFileName (NC.NoticedNoteFileNamesMenu NIL 'OPEN NC.NoteCardsIconWindow 'Close% NoteFile]
  (if NoteFileName
    then (NC.CloseNoteFile (NC.NoteFileFromFileName NoteFileName)
      NC.NoteCardsIconWindow)
    else (NC.PrintMsg NC.NoteCardsIconWindow NIL "Close cancelled." (CHARACTER 13))
      (DISMISS NC.MsgDelay)
      (NC.ClearMsg NC.NoteCardsIconWindow T))))
(|Close w/o confirm|
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu NIL 'OPEN NC.NoteCardsIconWindow 'Close% NoteFile]
    (if NoteFileName
      then (NC.CloseNoteFile (NC.NoteFileFromFileName NoteFileName)
        NC.NoteCardsIconWindow NIL T)
      else (NC.PrintMsg NC.NoteCardsIconWindow NIL "Close w/o confirm cancelled." (CHARACTER 13))
        (DISMISS NC.MsgDelay)
        (NC.ClearMsg NC.NoteCardsIconWindow T))))
(Abort% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu NIL 'OPEN NC.NoteCardsIconWindow 'Abort% NoteFile]
    (if NoteFileName
      then (NC.AbortSession (NC.NoteFileFromFileName NoteFileName)
        NC.NoteCardsIconWindow)
      else (NC.PrintMsg NC.NoteCardsIconWindow NIL "Abort cancelled." (CHARACTER 13))
        (DISMISS NC.MsgDelay)
        (NC.ClearMsg NC.NoteCardsIconWindow T))))
(Compact% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Compact% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.CompactNoteFile NoteFileName NIL NIL NC.NoteCardsIconWindow))))
(|Compact In Place|
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Compact% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.CompactNoteFile NoteFileName NIL T NC.NoteCardsIconWindow))))
(Inspect&Repair% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Inspect&Repair% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.InspectAndRepairNoteFile NoteFileName NIL NC.NoteCardsIconWindow))))
(Read% Substances
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Inspect&Repair% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.InspectAndRepairNoteFile NoteFileName T NC.NoteCardsIconWindow))))
(Copy% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow 'Copy% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.CopyNoteFile NoteFileName NIL NC.NoteCardsIconWindow))))
(Rename% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Rename% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.RenameNoteFile NoteFileName NIL NC.NoteCardsIconWindow))))
(Delete% NoteFile
  (LET [(NoteFileName (NC.NoticedNoteFileNamesMenu T 'CLOSED NC.NoteCardsIconWindow
    'Delete% NoteFile]
    (if (NULL NoteFileName)
      then NIL
      else (if (EQ NoteFileName 'NEW)
        then (SETQ NoteFileName NIL))
        (NC.DeleteDatabaseFile NoteFileName NC.NoteCardsIconWindow))))
(Create% NoteFile
  (NC.CreateDatabaseFile NIL NIL NIL NIL NIL NIL NC.NoteCardsIconWindow NIL NIL NIL NIL NIL NIL))
(|Create and Open|
  (NC.CreateDatabaseFile NIL NIL NIL NIL NIL NIL NC.NoteCardsIconWindow NIL NIL T NIL NIL NIL))
(LET ((NoteFileName (NC.NoticedNoteFileNamesMenu NIL 'EITHER NC.NoteCardsIconWindow)))
  (if NoteFileName
    then (APPLY* Op (NC.NoteFileFromFileName NoteFileName)
      NC.NoteCardsIconWindow))

```

**(NC.NoteCardsIconCursorEventFn**

[LAMBDA (Window)

(\* pmi%: "18-Mar-87 17:22")

(\* \* If not in an operation, then redisplay the icon window. The redisplay fn takes care of noticing whether the cursor is in or out of the icon.)

(\* \* pmi 3/18/87%: Uses the globalvar NC.NoteCardsIconWindow to solve problem of switching between the two session icon styles.)

```
(DECLARE (GLOBALVARS NC.NoteCardsIconWindow))
(if (NULL (WINDOWPROP NC.NoteCardsIconWindow 'NoteCardsOperationInProgress))
  then (REDISPLAYW NC.NoteCardsIconWindow])
```

**(NC.NoteCardsIconFn**

[LAMBDA (IconWindow Icon)

(\* pmi%: "18-Mar-87 17:22")

(\* \* Create the shrunken icon for the NoteCards session icon)

```
(DECLARE (GLOBALVARS NC.NoteCardsIconShrunkenBitmap NC.NoteCardsIconShrunkenMask))
(if Icon
  else (ICONW NC.NoteCardsIconShrunkenBitmap NC.NoteCardsIconShrunkenMask NIL T))
```

**(NC.ResetNoteCardsIcon**

[LAMBDA (Style)

(\* pmi%: "18-Mar-87 17:24")

(\* \* Called when switching between the two styles for the NoteCards session icon.  
If the requested style is already in use, just flashes the icon. Otherwise, closes the window, resets it, and brings up the new icon.)

```
(DECLARE (GLOBALVARS NC.NoteCardsIconWindow))
(if (NULL Style)
  elseif (EQ (WINDOWPROP NC.NoteCardsIconWindow 'NC.SessionIconStyle)
    Style)
  then (FLASHWINDOW NC.NoteCardsIconWindow)
  else (CLOSEW NC.NoteCardsIconWindow)
    (SETQ NC.NoteCardsIconWindow NIL)
    (NC.BringUpNoteCardsIcon NIL Style))
```

**(NC.MakeNewStyleNoteCardsIcon**

[LAMBDA (Position)

; Edited 20-Dec-88 15:26 by krivacic

;;; Creates a new style NoteCards Session Icon (designed by Peggy Irish)

;;; pmi 3/26/87: Added AFTERMOVEFNS to keep track of NC.NoteCardsIconPosition.

```
(DECLARE (GLOBALVARS NC.CardOpsMenu NC.NoteFileOpsMenu NC.OtherOpsMenu NC.CardOpsItems NC.NoteFileOpsItems
  NC.OtherOpsItems NC.NoteCardsIconBitmap NC.NoteCardsIconMask NC.NoteCardsIconPosition))
(LET ((WinWidth 115)
  (WinHeight 86)
  (MenuFont (FONTCREATE 'HELVETICA 10 'BOLD))
  WinPos IconWindow)
  [SETQ NC.NoteCardsIconPosition (SETQ WinPos (OR Position NC.NoteCardsIconPosition
    (GETBOXPOSITION WinWidth WinHeight NIL NIL NIL
      "Please position NoteCards icon."))
```

;;; Make the window an icon so that we can use the masking capability available for icons. This is what allows us to see the background around the icon,  
;;; instead of rectangular white space.

```
(SETQ IconWindow (ICONW NC.NoteCardsIconBitmap NC.NoteCardsIconMask WinPos T))
```

;;; Each menu item is actually a separate menu. The item regions needed to be offset horizontally, which is impossible within a menu.

```
(ADDMENU (create MENU
  ITEMS _ ' ("Card" NC.CardOpsMenu NC.CardOpsItems))
  CENTERFLG _ T
  MENUFONT _ MenuFont
  ITEMWIDTH _ 64
  MENUOUTLINESIZE _ 1)
IconWindow
(create POSITION
  XCOORD _ 12
  YCOORD _ 33)
T)
(ADDMENU (create MENU
  ITEMS _ ' ("NoteFile" NC.NoteFileOpsMenu NC.NoteFileOpsItems))
  CENTERFLG _ T
  MENUFONT _ MenuFont
  ITEMWIDTH _ 64
  MENUOUTLINESIZE _ 1)
IconWindow
(create POSITION
  XCOORD _ 20
  YCOORD _ 46)
```

```

T)
(ADDMENU (create MENU
  ITEMS _ ' ("Other" NC.OtherOpsMenu NC.OtherOpsItems))
  CENTERFLG _ T
  MENUFONT _ MenuFont
  ITEMWIDTH _ 64
  MENUOUTLINESIZE _ 1)
IconWindow
(create POSITION
  XCOORD _ 28
  YCOORD _ 59)
T)
(OPENW IconWindow)
(WINDOWPROP IconWindow 'ICONFN (FUNCTION NC.NoteCardsIconFn))
(WINDOWPROP IconWindow 'AFTERMOVEFN (FUNCTION NC.NoteCardsIconAfterMoveFn))
(WINDOWPROP IconWindow 'BUTTONEVENTFN (FUNCTION NC.NoteCardsIconButtonEventFn))
(WINDOWPROP IconWindow 'SCROLLFN NIL)
(WINDOWPROP IconWindow 'NOSCROLLBARS T)

```

;;; Need the RedisplayFn for icons

```

(WINDOWPROP IconWindow 'REPAINTFN (FUNCTION \ICONW.REPAINTFN))
(WINDOWPROP IconWindow 'CURSORINFN NIL)
(WINDOWPROP IconWindow 'CURSORMOVEDFN NIL)
(WINDOWPROP IconWindow 'RESHAPEFN 'DON'T)
(WINDOWPROP IconWindow 'NC.SessionIconStyle 'NewStyle)
IconWindow])

```

### (NC.MakeOldStyleNoteCardsIcon

[LAMBDA (Position)

(\* pmi%: "26-Mar-87 15:44")

(\* \* Creates an old style NoteCards Session Icon (designed by Frank Halasz))

(\* \* fgh |6/7/86| Added Position arg and NC.NoteCardsIconPosition globalvar reference.)

(\* \* fgh |6/24/86| Revamped for new 1.3 session icon. Session icon now a positional menu, but one with the buttoneventfn and cursormovedfns handled by NC and not the standard menu code. Menu shows only when cursor inside icon, rest of time the icon shows.)

(\* \* rht 8/28/86%: Now prevents reshaping of session icon.)

(\* \* pmi 3/19/87%: Separated code for Old style icon from code for New style icon. Most of this function was formerly NC.MakeNoteCardsIcon, which now calls this function for the Old style icon.)

(\* \* pmi 3/26/87%: Added AFTERMOVEFNS to keep track of NC.NoteCardsIconPosition.)

```

(DECLARE (GLOBALVARS NC.CardOpsMenu NC.NoteFileOpsMenu NC.OtherOpsMenu NC.CardOpsItems NC.NoteFileOpsItems
  NC.OtherOpsItems NC.OldStyleIconBitmap NC.OldStyleIconMenuBitmap NC.NoteCardsIconPosition))
(LET ((WinWidth (WIDTHIFWINDOW (BITMAPWIDTH NC.OldStyleIconBitmap)
  0))
  (WinHeight (HEIGHTIFWINDOW (BITMAPHEIGHT NC.OldStyleIconBitmap)
  NIL 0))
  WinPos IconWindow)
[SETQ NC.NoteCardsIconPosition (SETQ WinPos (OR Position NC.NoteCardsIconPosition
  (GETBOXPOSITION WinWidth WinHeight NIL NIL NIL
  "Please position NoteCards icon."))]
(SETQ IconWindow (ADDMENU (create MENU
  ITEMS _ ' ("Card Ops" NC.CardOpsMenu NC.CardOpsItems)
  ("NoteFile Ops" NC.NoteFileOpsMenu NC.NoteFileOpsItems)
  ("Other Ops" NC.OtherOpsMenu NC.OtherOpsItems))
  CENTERFLG _ T
  MENUFONT _ (FONTCREATE 'HELVETICA 12 'BOLD)
  MENUBORDERSIZE _ 1
  ITEMWIDTH _ (DIFFERENCE (BITMAPWIDTH NC.OldStyleIconBitmap)
  2)
  ITEMHEIGHT _ (FIX (TIMES 0.333333 (BITMAPHEIGHT NC.OldStyleIconBitmap)
  ))
  MENUOUTLINESIZE _ 1)
  (CREATEW (CREATEREGION (fetch (POSITION XCOORD) of WinPos)
  (fetch (POSITION YCOORD) of WinPos)
  WinWidth WinHeight)
  NIL 0 T)
  NIL T))
(OPENW IconWindow)
(BITBLT IconWindow 0 0 (SETQ NC.OldStyleIconMenuBitmap (BITMAPCREATE WinWidth WinHeight)))
(WINDOWPROP IconWindow 'ICONFN (FUNCTION NC.NoteCardsIconFn))
(WINDOWPROP IconWindow 'AFTERMOVEFN (FUNCTION NC.NoteCardsIconAfterMoveFn))
(WINDOWPROP IconWindow 'BUTTONEVENTFN (FUNCTION NC.NoteCardsIconButtonEventFn))
(WINDOWPROP IconWindow 'SCROLLFN NIL)
(WINDOWPROP IconWindow 'NOSCROLLBARS T)
(WINDOWPROP IconWindow 'REPAINTFN (FUNCTION NC.RedisplayNoteCardsIconWindow))
(WINDOWPROP IconWindow 'CURSORINFN (FUNCTION NC.NoteCardsIconCursorEventFn))
(WINDOWPROP IconWindow 'CURSOROUTFN (FUNCTION NC.NoteCardsIconCursorEventFn))
(WINDOWPROP IconWindow 'CURSORMOVEDFN NIL)
(WINDOWPROP IconWindow 'RESHAPEFN 'DON'T)
(WINDOWPROP IconWindow 'NC.SessionIconStyle 'OldStyle)

```

```
(BITBLT NC.OldStyleIconBitmap 0 0 IconWindow 0 0 WinWidth WinHeight 'INPUT 'REPLACE)
IconWindow])
```

```
)
```

```
;;; Command Menu interface to NC functions
```

```
(DEFINEQ
```

### (NC.CloseNoteCards

```
[LAMBDA (CardIdOrCardList NoCheckFlg DontClearFlg InterestedWindow)
(* Randy.Gobbel "2-Apr-87 15:38")

(* * Close note acrds on the screen)

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

(* * kirk 21Feb86 Added InterestedWindow)

(* * fgh |6/27/86| Fixed call to NC.SelectNoteCards to use just InterestedWindow)

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

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

(* * rht 3/9/87%: Changed NC.DeleteSelectingMenu to NC.SelectingCardsMenu.)

(* * rg |3/9/87| added NC.ProtectedSessionOperation wrapper)

(* * rg |4/1/87| changes NC.ProtectedSessionOperation to NCP.WithLockedCards ;
also added NC.IfAllCardsFree wrapper)

(DECLARE (GLOBALVARS NC.SelectingCardsMenu))
(NCP.WithLockedCards (NC.IfAllCardsFree (NC.LockListOfCards (MKLIST CardIdOrCardList)
"Close Note Cards")
(LET (Cards Window)
[SETQ Cards (COND
((LISTP CardIdOrCardList))
(CardIdOrCardList (NC.CoerceToCard CardIdOrCardList))
(T (NC.SelectNoteCards NIL NIL NC.SelectingCardsMenu
InterestedWindow "Please shift-select the cards to
be closed.")
(ALLOW.BUTTON.EVENTS)
(for Card in (MKLIST Cards) do (COND
((AND (NC.ActiveCardP Card)
(SETQ Window (NC.FetchWindow Card)))
(COND
((NEQ (NC.QuitCard Card T)
'DON'T)
(while (OPENWP Window)
do (BLOCK])
```

### (NC.CopyStructure

```
[LAMBDA (RootCards DestinationFileBox TraversalSpecs InterestedWindow QuietFlg)
(* pmi%: "29-Oct-87 17:17")

(* * Copy a NoteCard structure into a filebox)

(* * kirk 13/7/86%: Placed TraversalSpecs after RootCards selection and changed prompt message)

(* * rht 9/2/86%: Threw away CheckFlg arg. Wasn't being used.
Changed to call NCP.CollectCards instead of outdated NC.CollectCards.
Changed arg named ToPosition to DestinationFileBox. Also changed FromCard to RootCard.
Passes two link types to NC.AskTraversalSpecs.)

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

(* * rht 3/9/87%: Now accepts multiple root cards.)

(* * rg |3/9/87| added NC.ProtectedSessionOperation wrapper)

(* * RG |3/18/87| changed NC.ProtectedSessionOperation to NCP.WithLockedCards ;
added NC.IfAllCardsFree wrapper)

(* * rht&rg&pmi 4/22/87%: Removed calls to ERROR!)

(* * rg |6/2/87| added check for DON'T to selection of dest filebox)

(* * pmi 10/29/87%: Now saves cards returned from NCP.CollectCards to return to NC.MoveStructure.)

(DECLARE (GLOBALVARS NC.SelectingCardsMenu NC.SelectingCardMenu))
(NCP.WithLockedCards (SETQ RootCards (MKLIST RootCards))
(NC.IfAllCardsFree (NC.LockListOfCards RootCards "Copy Structure")
(LET (OriginalCards)
(if [AND (OR RootCards (SETQ RootCards (NC.SelectNoteCards NIL NIL NC.SelectingCardsMenu NIL
```

```

                                "Shift-select the root cards of the
                                structure"))
[OR TraversalSpecs (SETQ TraversalSpecs (NC.AskTraversalSpecs (fetch (Card
                                                                    NoteFile
                                                                    )
                                                                    of (CAR
                                                                    RootCards
                                                                    ))
                                                                    ' (SubBox FiledCard]
(OR DestinationFileBox (NEQ (SETQ DestinationFileBox
                             (NC.SelectNoteCards T [FUNCTION (LAMBDA (Card)
                             (NC.FileBoxP Card T]
                             NC.SelectingCardMenu NIL "Shift-select the
                             FileBox to contain the structure." T))
                             'DON'T])
then (NC.CopyCards (SETQ OriginalCards (NCP.CollectCards RootCards
                                         (fetch (TRAVERSALSPECS LinkTypes)
                                         of TraversalSpecs)
                                         (fetch (TRAVERSALSPECS Depth)
                                         of TraversalSpecs)))
         DestinationFileBox RootCards QuietFlg InterestedWindow)
      OriginalCards])

```

**(NC.CloseStructure**

```

[LAMBDA (RootCards TraversalSpecs InterestedWindow QuietFlg) (* Randy.Gobbel "2-Apr-87 15:38")

(* rht 9/2/86%: Replaced call to outdated NC.CollectCards with NCP.CollectCards.
Threw away useless NoCheckFlg and Don'tClearFlg args.)

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

(* rht 3/9/87%: Now accepts multiple root cards.)

(* rg [3/9/87] added NC.ProtectedSessionOperation wrapper)

(* rg [4/1/87] changes NC.ProtectedSessionOperation to NCP.WithLockedCards ;
added NC.IfAllCardsFree wrapper)

(NCP.WithLockedCards (SETQ RootCards (MKLIST RootCards))
  (NC.IfAllCardsFree (NC.LockListOfCards RootCards "Close Structure")
    (OR RootCards (SETQ RootCards (NC.SelectNoteCards NIL NIL NC.SelectingCardsMenu NIL
                                                "Shift-select the root cards of the structure")))
    (ERROR!))
  [OR TraversalSpecs (SETQ TraversalSpecs (NC.AskTraversalSpecs (fetch (Card NoteFile)
                                                                    of (CAR RootCards))
                                                                    ' (SubBox FiledCard]
    (if (AND RootCards TraversalSpecs)
      then (OR QuietFlg (NC.PrintMsg InterestedWindow T "Collecting cards to close ..."))
            (NC.CloseNoteCards (NCP.CollectCards RootCards (fetch (TRAVERSALSPECS LinkTypes)
                                                                    of TraversalSpecs)
                                                                    (fetch (TRAVERSALSPECS Depth) of TraversalSpecs))
                                NIL NIL InterestedWindow)
            (OR QuietFlg (NC.ClearMsg InterestedWindow T))
            RootCards])

```

**(NC.DeleteStructure**

```

[LAMBDA (RootCards TraversalSpecs InterestedWindow QuietFlg Don'tPutToBeDeletedCardsFlg)
(* Randy.Gobbel "2-Apr-87 15:38")

(* rht 8/29/86%: Reorganized and changed to call NCP.CollectCards which is more efficient than the old
NCP.ComputeTransitiveClosure. Also now takes QuietFlg and Don'tPutToBeDeletedCardsFlg args.
Threw away Don'tClearFlg.)

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

(* rht 3/9/87%: Now accepts multiple root cards.)

(* rg [3/9/87] added NC.ProtectedSessionOperation wrapper)

(* rg [4/2/87] turned NC.ProtectedSessionOperation into NC.CardSelectionOperation)

(NCP.WithLockedCards (SETQ RootCards (MKLIST RootCards))
  (NC.IfAllCardsFree (NC.LockListOfCards RootCards "Delete Structure")
    (OR RootCards (SETQ RootCards (NC.SelectNoteCards NIL NIL NC.SelectingCardsMenu NIL
                                                "Shift-select the root cards of the structure")))
    (ERROR!))
  [OR TraversalSpecs (SETQ TraversalSpecs (NC.AskTraversalSpecs (fetch (Card NoteFile)
                                                                    of (CAR RootCards))
                                                                    ' (SubBox FiledCard]
    (if (AND RootCards TraversalSpecs)
      then (OR QuietFlg (NC.PrintMsg InterestedWindow T "Collecting cards to delete ..."))
            (NC.DeleteNoteCards (NCP.CollectCards RootCards (fetch (TRAVERSALSPECS LinkTypes)
                                                                    of TraversalSpecs)
                                                                    (fetch (TRAVERSALSPECS Depth) of TraversalSpecs))
                                T NIL InterestedWindow QuietFlg NIL Don'tPutToBeDeletedCardsFlg)

```



```
(OR QuietFlg (NC.ClearMsg InterestedWindow T))
RootCards])
```

**(NC.MoveStructure**

```
[LAMBDA (RootCards DestinationFileBox TraversalSpecs InterestedWindow QuietFlg Don'tPutToBeDeletedCardsFlg)
(* Randy.Gobbel "2-Jun-87 19:44")
```

```
(* * Copy a NoteCard structure into a filebox)
```

```
(* * rht 9/2/86%: Added QuietFlg and Don'tPutToBeDeletedCardsFlg args.
Changed names of a few args and removed Don'tClearFlg arg.
Took out REVERSE to save time and space.)
```

```
(* * rht 3/9/87%: Now accepts multiple root cards.)
```

```
(* * rht&rg&pmi 4/22/87%: Now checks that NC.CopyStructure returns valid stuff before deleting.)
```

```
(* * rg [6/2/87] added NCP.WithLockedCards wrapper)
```

```
(NCP.WithLockedCards (MKLIST RootCards)
(NC.IfAllCardsFree (NC.LockListOfCards RootCards "Move Structure")
(LET ((Structure (NC.CopyStructure (MKLIST RootCards)
DestinationFileBox TraversalSpecs InterestedWindow QuietFlg)))
(AND Structure (NC.DeleteNoteCards Structure T NIL InterestedWindow QuietFlg NIL
Don'tPutToBeDeletedCardsFlg])
```

```
)
```

```
;;; NoteFile interface
```

```
(DEFINEQ
```

**(NC.NoteFileMenuWindowP**

```
[LAMBDA (Window) ; Edited 19-Feb-88 12:27 by Randy.Gobbel
;; IS Window a NoteFile menu?
;; fgh 6/13/86 First created.
(AND Window (WINDOWPROP Window 'NoteFile]))
```

**(NC.SetUpNoteFileInterface**

```
[LAMBDA (NoteFile Position InterestedWindow Don'tCreateInterfaceFlg)
; Edited 7-Dec-88 17:21 by krivacic
```

```
;;; Create the NoteCards control menu for a NoteFile
```

```
;; kirk 13Jan85 Decreased the size of the NoteFile Menu
;; fgh 1/22/86 Fixed the ghost box size when position menu.
;; rht 5/6/86: Now restores the menu's WhenSelectedFn and ungrays its items if already existed.
;; fgh 6/27/86 Added position argument
;; rht 7/5/86: Now shades NewCards if readonly notefile.
;; fgh 7/6/86 Will now set up menu correctly even if NF is closed.
;; rht 7/13/86: Was ignoring the Position arg. No longer.
;; rht 11/20/86: Changed name from ShowBox to ShowCards.
;; pmi 12/12/86: Added InterestedWindow argument so that we can print a prompt to the user about placing a newly created NoteFile menu.
;; pmi 3/20/87: Removed WhenSelectedFn when overhauling to have NewCards and ShowCards middlebutton menus appear when buttoned
;; DOWN, instead of after the button comes back up. Everything is now done in the ButtonEventFn.
;; pmi 5/6/87: Moved prompt for position of icon to better place. Also added MENUOFFSET to NoteFile menu for Lyric.
;; pmi 5/19/87: Now stores the menu as a property of the fullfilename of the notefile. We might lose our pointer to the notefile object if another one
;; gets created with the same UID, but we would like to keep a pointer to the menu.
;; pmi 5/28/87: Now returns the NoteFile Interface window.
;; rg 12/15/87: fixes up menu grid so redisplay works when notefile is reopened.
;; pmi 12/17/87 Added Don'tCreateInterfaceFlg argument in response to suggestion by dsj. Now can be called to update the notefile interface and
;; won't automatically create a new one if it doesn't already exist.
;; pmi 1/11/88: moved rg's last patch so that it applies to both new and preexisting notefile icon menus.
;; pmi 8/29/88: No longer tries to set windowprops on NIL window, which were ending up on the Mouse TTY window.
;; pmi 9/12/88: Now uses NCP.GrayShade instead of GRAYSHADE for shading menu items.
;; pmi 9/19/88: Now shades text of disabled menu items on notefile icons, instead of their backgrounds.
```

```
(DECLARE (GLOBALVARS NCP.GrayShade))
(LET ((Font (FONTCREATE 'HELVETICA 10 'BOLD))
(TitleFont (FONTCREATE 'HELVETICA 12 'BOLD))
NoteFileMenuWindow NoteFileMenu FullFileName) ; Main Menu
(SETQ FullFileName (fetch (NoteFile FullFileName) of NoteFile))
[if (SETQ NoteFileMenu (OR (NC.GetNoteFileMenu NoteFile)
(NC.GetNoteFileMenu FullFileName)))
then [replace (MENU TITLE) of NoteFileMenu with (CONCAT (if (NC.ReadOnlyNoteFileP NoteFile)
```

[illegible]

```

        (NC.MoveWindowOntoScreen NoteFileMenuWindow))
    (replace (NoteFile Menu) of NoteFile with NoteFileMenu)
    (PUTPROP FullFileName 'Menu NoteFileMenu)
    NoteFileMenuWindow])

        (if (NC.NoteFileOpenP
              (WINDOWPROP Window 'NoteFile))
            then
              (NC.PrintMsg Window T "Cannot Close
                                    Window While NoteFile is
                                    still open.")
              (FLASHWINDOW Window 3)
              [CLOSEW (CAR (WINDOWPROP
                            Window
                            'PROMPTWINDOW)
                        'DON'T])

```

**(NC.ResetNoteFileInterface**

[LAMBDA (NoteFile)

; Edited 3-Jan-89 17:28 by sye

;; Close up the control menu for a NoteFile

```

;; rht 5/1/86: Now does nothing if NC.NoteFileMenuLingerFlg is non-nil.
;; rht 5/6/86: Now smashes WhenSelectedFn and shades menu items if NC.NoteFileMenuLingerFlg is non-nil.
;; pmi 2/18/87: Now deletes 'RO:' in title of NoteFile menu if NoteFile was open Read-only
;; pmi 3/20/87: Removed WhenSelectedFn when overhauling to have NewCards and ShowCards middlebutton menus appear when buttoned
;; DOWN, instead of after the button comes back up. Everything is now done in the ButtonEventFn.
;; pmi 5/1/87: Inserted call to NC.ClearMsg before the NoteFile's icon is deleted. Added Window arg to DELETEMENU call. Also commented out
;; the code that deletes the NoteFile's menu and removes the NoteFile from its window if NC.NoteFileMenuLingerFlg is NIL. That way if the
;; NoteFile is reopened, it will remember the position for its icon.
;; pmi 9/12/88: Now uses NCP.GrayShade instead of GRAYSHADE for shading menu items.
;; pmi 9/19/88: Now shades text of disabled menu items on notefile icons, instead of their backgrounds.
;; sye 12/22/88: Detach FileCapacity window
(DECLARE (GLOBALVARS NC.NoteFileMenuLingerFlg NCP.GrayShade))
(LET ((Menu (fetch (NoteFile Menu) of NoteFile))
      Window FullFileName)
  (if Menu
    then (SETQ Window (WFROMMENU Menu))
          (DETACHALLWINDOWS Window)
          (WINDOWPROP Window 'FileCapacityFlag NIL)
          (if NC.NoteFileMenuLingerFlg
            then [for Item in (fetch (MENU ITEMS) of Menu)
                  do (if (NC.NoteFileOpenP NoteFile)
                        then (SHADEITEM Item Menu WHITESHADE)
                        else (NEWSHADEITEM Item Menu NCP.GrayShade NIL 'ERASE)]
            (if (NC.ReadOnlyNoteFileP NoteFile)
              then (SETQ FullFileName (fetch (NoteFile FullFileName) of NoteFile))
                   [replace (MENU TITLE) of Menu with (CONCAT (FILENAMEFIELD FullFileName
                                                                              'NAME)
                                                                "; "
                                                                (FILENAMEFIELD FullFileName
                                                                              'VERSION))

                   (replace (MENU IMAGE) of Menu with NIL)

                   ;; KLUDGE. Fetching the image height forces the menu package to recompute the menu image.
                   (fetch (MENU IMAGEHEIGHT) of Menu)

                   ;; further KLUDGE. Have to smash MENUGRID to prevent menu image from being shifted over.
                   (replace (MENU MENUGRID) of Menu with (LIST 2 2 (fetch (MENU ITEMWIDTH)
                                                                           of Menu)
                                                                (fetch (MENU ITEMHEIGHT) of Menu)))

                   (for Item in (fetch (MENU ITEMS) of Menu) when (EQ (CAR Item)
                                                                           'NewCards)
                     do (NEWSHADEITEM Item Menu NCP.GrayShade NIL 'ERASE))
                   (AND Window (REDISPLAYW Window)))
          else (NC.ClearMsg Window T)
                (WINDOWPROP Window 'CLOSEFN NIL)
                (DELETEMENU Menu T Window])

```

**(NC.NoteFileIconButtonEventFn**

[LAMBDA (Window)

; Edited 7-Dec-88 17:20 by krivacic

```

;; Bring up NoteFile Menues
;; kirk 15Jul86 Adjusted title size check for change in font
;; rht 11/23/86: Now calls NC.NoteFileIconMiddleButtonFn to put up a menu of middle button options.
;; pmi 3/20/87: Overhauled to have NewCards and ShowCards middlebutton menus appear when buttoned DOWN, instead of after the button
;; comes back up.
;; pmi 4/3/87: Now unshades NewCards item during card creation to indicate that multiple cards may be created at the same time.
;; rg 11/4/87 calls NC.EditNoteCard w/ ReadOnly if NoteFile is read-only
;; pmi 2/23/88: Now passes Window as InterestedWindow argument to NC.MakeNoteCard.

```

;; pmi 9/12/88: Now uses NCP.GrayShade instead of GRAYSHADE to indicate the disabled state of a menu item. Also uses BLACKSHADE instead of GRAYSHADE to shade the menu items when they are selected.

;; pmi 9/19/88: Now shades text of disabled menu items on notefile icons, instead of their backgrounds.

```
(DECLARE (GLOBALVARS BLACKSHADE NCP.GrayShade))
(LET
  (NoteFile Menu)
  (if (MOUSESTATE UP)
    else
      (TOTOPW Window)
      (LET [(Menu (CAR (WINDOWPROP Window 'MENU) ; title bar
        (SETQ NoteFile (WINDOWPROP Window 'NoteFile))
        (for Item in (fetch (MENU ITEMS) of Menu) when (INSIDEP (MENUITEMREGION Item Menu)
          (LASTMOUSEX Window)
          (LASTMOUSEY Window))
        do (RESETLST
          (RESETSAVE (SHADEITEM Item Menu BLACKSHADE)
            (if (NULL (NC.NoteFileOpenP NoteFile))
              then (LIST 'NEWSHADEITEM Item Menu NCP.GrayShade NIL 'ERASE)
              elseif (AND (EQ (CAR Item)
                'NewCards)
                (NC.ReadOnlyNoteFileP NoteFile))
              then (LIST 'NEWSHADEITEM Item Menu NCP.GrayShade NIL 'ERASE)
              else (LIST 'SHADEITEM Item Menu WHITESHADE)))
            (if (NULL (NC.NoteFileOpenP NoteFile))
              then
```

;; If the NoteFile is not open, just print a message and return

```
(FLASHW Window)
(NC.PrintMsg Window T (fetch (NoteFile FullFileName) of NoteFile)
  " is not an open notefile.")
(DISMISS 1000)
(NC.ClearMsg Window T)
elseif (PROCESSP (NC.NoteFileProp NoteFile 'ProcessInProgress))
  then
```

;; This will probably be overhauled with Randy G.'s concurrency fixes

```
(NC.PrintOperationInProgressMsg Window (CAR Item)
  (NC.NoteFileProp NoteFile 'OperationInProgress))
NIL
else (SELECTQ (CAR Item)
  (New% Cards (if (NC.ReadOnlyNoteFileP NoteFile)
    then (FLASHW Window)
    (NC.PrintMsg Window T "Can't create a new card in a
      Read-Only notefile.")
    (DISMISS 1000)
    (NC.ClearMsg Window T)
    elseif (LASTMOUSESTATE LEFT)
    then ;; For the left button, don't do anything until the button comes back up.
      ;; Otherwise, things happen too soon.
      (UNTILMOUSESTATE UP)
      (if (INSIDEP (MENUITEMREGION Item Menu)
        (LASTMOUSEX Window)
        (LASTMOUSEY Window))
        then (SHADEITEM Item Menu WHITESHADE)
        (NC.MakeNoteCard NC.DefaultCardType NoteFile NIL
          NIL NIL NIL Window))
      else (SHADEITEM Item Menu WHITESHADE)
      (NC.MakeNoteCard NIL NoteFile NIL NIL NIL NIL Window)))
  (Special% Cards
    (if (LASTMOUSESTATE LEFT)
      then ;; For the left button, don't do anything until the button comes back up. Otherwise, things
        ;; happen too soon.
        (UNTILMOUSESTATE UP)
        (if (INSIDEP (MENUITEMREGION Item Menu)
          (LASTMOUSEX Window)
          (LASTMOUSEY Window))
          then (NC.EditNoteCard (fetch (NoteFile TableOfContentsCard)
            of NoteFile)
            (fetch (NoteFile ReadOnlyFlg) of NoteFile)))
          else (NC.ChooseTopLevelCard NoteFile)))
      NIL)))
  (RETURN)
  finally (if (LASTMOUSESTATE MIDDLE)
    then (NC.NoteFileIconButtonMiddleButtonFn Window NoteFile)
    elseif (NULL (WINDOWPROP Window 'BusyOperation))
    then (RESETLST
      (RESETSAVE (WINDOWPROP Window 'BusyOperation "Operation")
        (WINDOWPROP Window BusyOperation NIL))
      (NC.NoteFileOperations NoteFile))
    else (NC.PrintMsg NIL NIL (CHARACTER 13)
      (WINDOWPROP Window 'BusyOperation)
      " in progress. Please wait.")]
```

**(NC.NoteFileOperations**

[LAMBDA (NoteFile)

(\* Randy.Gobbel "12-Mar-87 17:51")  
 (\* Ask user to choose which database operation.  
 Called from main menu whenselected fn.)

(\* fgh |11/14/85| Updated to handle NoteFile object and per NoteFile menus.)  
 (\* kirk 14Jan86 Added Close/Delete cards and changed the format)  
 (\* fgh |6/13/86| Now sets BusyOperation of NF menu to something interesting.)  
 (\* kirk 3Jul86 passed NoteFileMenuWindow to NC.CompactNoteFile)  
 (\* fgh |7/5/86| Fixed above fix and put NoteFileMenuWindow is all calls.)  
 (\* rht 7/5/86%: Added read-only open to notefileops menu.)  
 (\* kirk |7/14/86| Took comma out of FullFileName in NC.DeleteDatabaseFile call)  
 (\* kirk 15Jul86 Added Grey shade to non-functional items when file is open/closed)  
 (\* kef 8/6/86%: Added SPAWN.MOUSE towards the end to eliminate the extraneous printing of "NoteFile Busy" which seemed to be constantly occurring for remote NoteFiles.)  
 (\* pmi |12/4/86| Removed menu creation and shading of operations for open or closed NoteFile.  
 Now just uses appropriate pre-defined menu, depending on whether NoteFile is open or closed.)  
 (\* rht 2/11/87%: Now makes sure NC.OpenedNoteFileMenu and NC.ClosedNoteFileMenu are non-nil.)

```
(DECLARE (GLOBALVARS NC.OpenedNoteFileMenu NC.ClosedNoteFileMenu))
(if (OR (NULL NC.OpenedNoteFileMenu)
        (NULL NC.ClosedNoteFileMenu))
    then (NC.MakeNoteFileIconOperationsMenus) )
(LET (Menu)
  (if (NC.NoteFileOpenP NoteFile)
      then (SETQ Menu NC.OpenedNoteFileMenu)
      else (SETQ Menu NC.ClosedNoteFileMenu))
  (PUTMENUPROP Menu 'NoteFile NoteFile)
  (ALLOW.BUTTON.EVENTS)
  (MENU Menu))
```

**(NC.NoteFileIconMiddleButtonFn**

[LAMBDA (NoteFileIconWindow NoteFile)

(\* pmi%: " 1-Apr-87 15:35")

(\* Bring up the middle button menu for notefile icons.)  
 (\* pmi 4/1/87%: Disabled storing the middlebutton menu on the NoteFileIconWindow, so that it will always recompute to handle changes in the menu font.)

```
(DECLARE (GLOBALVARS NC.MenuFont NC.DefaultNoteFileIconMiddleButtonItem))
(LET (Menu)

  (* (OR (SETQ Menu (WINDOWPROP NoteFileIconWindow (QUOTE NoteFileMiddleButtonMenu)))
        (WINDOWPROP NoteFileIconWindow (QUOTE NoteFileMiddleButtonMenu))
        (SETQ Menu (create MENU ITEMS_ (APPEND NC.DefaultNoteFileIconMiddleButtonItem
        (NC.NoteFileProp NoteFile (QUOTE NoteFileIconMiddleButtonItem))) CHANGEOFFSETFLG
        (QUOTE Y) MENUOFFSET_ (CONS -1 0) CENTERFLG_ NIL MENUTITLEFONT_ NC.MenuFont MENUFONT_
        NC.MenuFont ITEMHEIGHT_ (ADD1 (FONTPROP NC.MenuFont 'HEIGHT)))
        (QUOTE HEIGHT))) WHENSELECTEDFN_ (FUNCTION (LAMBDA
        (Item Menu Button) (APPLY* (CADR Item) (GETMENUPROP Menu
        (QUOTE NoteFile) Item))))))

  [SETQ Menu (create MENU
    ITEMS_ (APPEND NC.DefaultNoteFileIconMiddleButtonItem (NC.NoteFileProp NoteFile
    NoteFileIconMiddleButtonItem))
    CHANGEOFFSETFLG_ 'Y
    MENUOFFSET_ (CONS -1 0)
    CENTERFLG_ NIL
    MENUFONT_ NC.MenuFont
    ITEMHEIGHT_ (ADD1 (FONTPROP NC.MenuFont 'HEIGHT))
    WHENSELECTEDFN_ (FUNCTION (LAMBDA (Item Menu Button)
      (APPLY* (CADR Item)
        (GETMENUPROP Menu 'NoteFile)
        Item))
    (PUTMENUPROP Menu 'NoteFile NoteFile)
    (SPAWN.MOUSE)
    (MENU Menu))
```

**(NC.PopUpNoteFileName**

[LAMBDA (NoteFile Item)

(\* pmi%: " 1-Apr-87 17:14")

(\* \* Popup the notefile's full name. Use menu stashed on NoteFileIconWindow if any.)

(\* \* pmi 4/1/87%: Added NC.MenuFont to all menus. Also disabled storing the filename menu on the NoteFileIconWindow, so that it will always recompute to handle changes in the menu font.)

```
(DECLARE (GLOBALVARS NC.MenuFont))
(LET [(NoteFileIconWindow (FROMMENU (fetch (NoteFile Menu) of NoteFile)

(* (MENU (OR (WINDOWPROP NoteFileIconWindow (QUOTE NameMenu))
(LET ((Menu (create MENU ITEMS _ (LIST (LIST (fetch (NoteFile FullFileName) of NoteFile) NIL
(fetch (NoteFile FullFileName) of NoteFile))) TITLE _ NIL MENUFONT _ NC.MenuFont)))
(WINDOWPROP NoteFileIconWindow (QUOTE NameMenu) Menu) Menu))))

(MENU (create MENU
ITEMS _ (LIST (LIST (fetch (NoteFile FullFileName) of NoteFile)
NIL
(fetch (NoteFile FullFileName) of NoteFile)))
TITLE _ NIL
MENUFONT _ NC.MenuFont
ITEMHEIGHT _ (PLUS (FONTPROP NC.MenuFont 'HEIGHT)
2))
```

## (NC.DisplayFileCapacity

```
[LAMBDA (NoteFile Item Caller)
(DECLARE (GLOBALVARS NC.MenuFont)) ; Edited 4-Jan-89 09:29 by sye
(LET* ((NoteFileIconW (FROMMENU (fetch (NoteFile Menu) of NoteFile)))
(Flag (WINDOWPROP NoteFileIconW 'FileCapacityFlag))
(IndexSize (fetch (NoteFile HashArraySize) of NoteFile))
[NumUsed (SUB1 (DIFFERENCE (fetch (NoteFile NextIndexNum) of NoteFile)
(LENGTH (fetch (NoteFile IndexNumsFreeList) of NoteFile))
(LineCount 1)
(Threshold 90)
(DisplayP T)
NameToPrint YSize SameYSize PercentageUsed AttachedWindows FileCapacityW)
[if (NULL Caller)
then (if (NULL (fetch (NoteFile Stream) of NoteFile))
then (NC.PrintMsg NIL T "Note File is closed")
(SETQ DisplayP NIL))
else (if (EQ Caller 'NC.GetNewCard)
then (if (OR (EQ Flag 'STAR)
(AND (NULL Flag)
(LESSP (QUOTIENT (TIMES 100 NumUsed)
IndexSize)
Threshold)))
then (SETQ DisplayP NIL]
(if DisplayP
then (SETQ PercentageUsed (QUOTIENT (TIMES 100 NumUsed)
IndexSize))
(SETQ AttachedWindows (ATTACHEDWINDOWS NoteFileIconW))
(SETQ NameToPrint (CONCAT PercentageUsed "% Full (" NumUsed "/" IndexSize ")))
(if (AND (GEQ PercentageUsed Threshold)
(NULL (fetch (NoteFile ReadOnlyFlag) of NoteFile)))
then (SETQ NameToPrint (CONCAT NameToPrint (CHARACTER 13)
"Compact NoteFile!"))
(SETQ LineCount 2))
(SETQ YSize (TIMES LineCount (NC.NoteFileIndicatorWinHeight)))
(if [SETQ FileCapacityW (for window in AttachedWindows thereis (WINDOWPROP window
'NoteFileCapacity)
then (SETQ SameYSize (EQ (fetch (REGION HEIGHT) of (WINDOWPROP FileCapacityW 'REGION))
YSize))
(if (AND FileCapacityW SameYSize)
then (CLEARW FileCapacityW)
(PRIN1 NameToPrint FileCapacityW)
else (AND FileCapacityW (REMOVEWINDOW FileCapacityW))
(NC.WithTopWindowsUnattached NoteFileIconW (SETQ FileCapacityW
(CREATEW (CREATEREGION 0 0 100 YSize)
NIL T))
(ATTACHWINDOW FileCapacityW NoteFileIconW 'TOP 'JUSTIFY 'LOCALCLOSE)
(WINDOWADDPROP FileCapacityW 'CLOSEFN [FUNCTION (LAMBDA (Window)
(LET
((PercentageUsed
(WINDOWPROP Window
'PercentageUsed))
(Threshold
(WINDOWPROP Window
'Threshold))
(MainW (MAINWINDOW Window))
NewFlag)
SETQ NewFlag
(if (GEQ PercentageUsed
Threshold)
then 'STAR))
(WINDOWPROP MainW
'FileCapacityFlag
NewFlag]
```

```

T)
[WINDOWPROP FileCapacityW 'REPAINTFN (FUNCTION (LAMBDA (Window)
(NC.DisplayFileCapacity
(WINDOWPROP Window
'NoteFile)
NIL
'REPAINTFN]
(WINDOWPROP FileCapacityW 'NoteFileCapacity T)
(WINDOWPROP FileCapacityW 'NoteFile NoteFile)
(WINDOWPROP FileCapacityW 'PercentageUsed PercentageUsed)
(WINDOWPROP FileCapacityW 'Threshold Threshold)
(WINDOWPROP NoteFileIconW 'FileCapacityFlag 'OPEN)
(DSPFONT NC.NoteFileIndicatorFont FileCapacityW)
(PRIN1 NameToPrint FileCapacityW)
(OPENW FileCapacityW))
)
(RPAQ? NC.DefaultNoteFileIconMiddleButtonItems '((|Full File Name| NC.PopUpNoteFileName "Display full file name of
this notefile.")
(File% Capacity NC.DisplayFileCapacity "Display file capacity of
this notefile")))

;;; Show NoteFile Indicators
(DEFINEQ
(NC.ShowNoteFiles
[LAMBDA (OffFlg)
(* kirk%: "28-May-86 14:41")
(* * Attach a NoteFile indicator to all Cards on the screen. Detach if OffFlg is true.)
(for window in (OPENWINDOWS) do (NC.AttachNoteFileName window OffFlg])
(NC.AttachNoteFileName
[LAMBDA (CardWindow OffFlg)
(* rht%: "25-Mar-87 14:46")
(* * Attach an indicator of the NoteFile for this Card to the Card)
(* * fgh |5/29/86| Changed collection of old attached windows to prevent NILs in collection result.)
(* * fgh |6/13/86| Put detachment and reattachment of previous attached windows into a RESETLST.
Added FREEATTACHEDWINDOW as the CLOSEFN of the indicator window.
Added code to reset PASSTOMAINWINDOW COMS of the previously attached windows.
Changed MINSIZE of indicator window to allow small fonts. Added a REPAINTFN to the indicator window.)
(* * fgh |7/2/86| Replaced the CLEARW call in the indicator window REPAINTFN with a DSPRESET for two reasons%: reset
the X coord before printing and CLEARW is not allowed in REPAINTFNS.)
(* * fgh&rht 7/5/86%: Changed the name "FileName" to "NameToPrint" %.)
(* * rht 3/20/87%: Moved Frank's hack that temporarily detaches windows into a macro called NC.WithWindowsUnattached.)
(* * rht 3/25/87%: Now calls NC.NoteFileIndicatorWinHeight.)
(LET ((Card (NC.CardFromWindow CardWindow))
AttachedWindows FileName Window)
(if (NC.CardP Card)
then (SETQ AttachedWindows (ATTACHEDWINDOWS CardWindow))
(if OffFlg
then [CLOSEW (for window in AttachedWindows thereis (WINDOWPROP window 'NoteFileIndicator]
(REPOSITIONATTACHEDWINDOWS CardWindow)
else (TOTOPW CardWindow)
(if [SETQ Window (for window in AttachedWindows thereis (WINDOWPROP window
'NoteFileIndicator]
then
(* * There already is an indicator window, open and redisplay it.)
(OR (OPENWP Window)
(OPENW Window))
(REDISPLAYW Window)
else
(* * No existing indicator window, make one.)
(NC.WithTopWindowsUnattached CardWindow)
(* * Make the indicator window.)
(LET (NoteFile NameToPrint YSize IndicatorWindow)
(SETQ NoteFile (fetch (Card NoteFile) of Card))
(SETQ NameToPrint (NC.MakeNoteFileNameIndicator NoteFile))
(SETQ YSize (NC.NoteFileIndicatorWinHeight))
(SETQ IndicatorWindow (CREATEW (CREATEREGION 0 0 100 YSize)

```

```
(RPAQ? NC.FileBrowserMenuItemsToAdd
'((Open NC.FileBrowserOpen "Open selected Notefiles." (SUBITEMS (Open% Read/Write NC.FileBrowserOpen
                                "Opens a notefile for Read/Write
                                access.")
                                (Read-only% Open (NC.FileBrowserOpen INPUT)
                                "Open selected Notefiles for
                                read-only.))))
(Checkpoint NC.FileBrowserCheckpoint "Checkpoint the selected Notefiles, saving dirty cards.")
(Close NC.FileBrowserClose "Close selected Notefiles.")
(Abort NC.FileBrowserAbort "Aborts the selected Notefiles losing work since last checkpoint.")
(Compact NC.FileBrowserCompact "Compacts selected Notefiles copying to new files."
  (SUBITEMS (|Compact To New File| NC.FileBrowserCompact "Compacts selected Notefiles copying to
              new files.")
  (|Compact In Place| (NC.FileBrowserCompact 'InPlace)
    "Compacts selected Notefiles w/o copyig to new files.))))
(Inspect&Repair NC.FileBrowserInspect&Repair "Inspects and optionally repairs selected Notefiles."
  (SUBITEMS (From% Links NC.FileBrowserInspect&Repair "Do an Inspect & Repair on a NoteFile from
              the Link information.")
  (From% Contents (NC.FileBrowserInspect&Repair 'ReadSubstances)
```



```

        "Inspects and optionally repairs selected Notefiles, but reads every substance.
        This slows things WAY down.)))
    (Copy FB.COPYCOMMAND "Copies given notefile(s) to target file (or directory).")
    (Rename FB.RENAMECOMMAND "Moves given notefile(s) to target file (or directory)."))))

```

```

(RPAQ? NC.FileBrowserExpungeMenuItem ' (Expunge NC.FileBrowserExpunge "Permanently removes from the file system all
files marked for deletion"))

```

```

(RPAQ? NC.FileBrowserRecomputeMenuItem ' (Recompute FB.UPDATECOMMAND "Recomputes set of notefiles satisfying
selection pattern" (SUBITEMS ("Same Pattern" FB.UPDATECOMMAND
                                "Recomputes set of files
                                satisfying current pattern")
                                ("New Pattern"
                                NC.FileBrowserRecomputeWithNewPattern
                                "Prompts for a new selection
                                pattern and updates browser")
                                ("New Info" FB.NEWINFOCOMMAND "Change the
                                set of file attributes that are
                                displayed"))))

```

;;; Build the menu items for notecards file browser.

```

(RPAQ? NC.FileBrowserMenuItems
  (APPEND NC.FileBrowserMenuItemsToAdd (for MenuItem in FB.MENU.ITEMS when (FMEMB (CAR MenuItem)
                                                                                     NC.FileBrowserMenuItemsToKeep)
                                         collect
                                         (if (EQ (CAR MenuItem)
                                             'Recompute)
                                             then NC.FileBrowserRecomputeMenuItem else MenuItem))
        (LIST NC.FileBrowserExpungeMenuItem)
        (for MenuItem in FB.MENU.ITEMS unless (OR (FMEMB (CAR MenuItem)
                                                           NC.FileBrowserMenuItemsToRemove)
                                                    (FMEMB (CAR MenuItem)
                                                           NC.FileBrowserMenuItemsToKeep))
          collect MenuItem)))

```

```

(RPAQ? NC.FileBrowserPatterns NIL)

```

```

(RPAQ? NC.FileBrowserDefaultPatterns ' ({DSK}/HOME/IL/NOTECARDS/NOTEFILES/*.NOTEFILE;* {DSK}*.NOTEFILE;*))

```

;;; Functions called from file browser menu.

```

(DEFINEQ

```

```

(NC.FileBrowserOpen

```

```

  [LAMBDA (Browser Key Item Menu Access)

```

```

; Edited 27-Feb-89 18:13 by krivacic

```

;;; Function called from file browser menu for notefile open.

;;; rht 7/4/86: Added Access arg.

;;; rht 3/14/87: Changed to call NC.OpenNoteFile.

;;; pmi 5/20/87: Removed HashArray argument in calls to NC.OpenNoteFile.

;;; bk 2/27/89: Add NC.BringUpTOC param to NC.OpenNoteFile.

```

  (for FileObject in (FB.SELECTEDFILES Browser) bind FileName
    do (NC.OpenNoteFile [SETQ FileName (U-CASE (MKATOM (FB.FETCHFILENAME FileObject)
                                                         (NoteFile (NC.NoteFileFromFileName FileName)))
                    NIL NIL NIL NIL NIL NIL NIL (if (NOT (NC.NoteFileFromFileName FileName))
                                                       then (fetch (FILEBROWSER PROMPTWINDOW) of Browser))
                    NIL NIL NIL Access NIL NIL NC.BringUpTOC])

```

```

(NC.FileBrowserClose

```

```

  [LAMBDA (Browser Key Item Menu)

```

```

(* rht%: " 2-Jul-86 18:47")

```

```

(* * Function called from file browser menu for notefile close.)

```

```

(* * rht 7/2/86%: Now passes proper InterestedWindow to NC.CloseDatabaseFile.
Also calls FB.PROMPTWPRINT instead of NC.PrintMsg.)

```

```

  (for FileObject in (FB.SELECTEDFILES Browser) do (LET* ((FileName (MKATOM (FB.FETCHFILENAME FileObject)))
                                                           (NoteFile (NC.NoteFileFromFileName FileName)))
    (if (type? NoteFile NoteFile)
        then (NC.CloseDatabaseFile NoteFile
                                     (fetch (FILEBROWSER PROMPTWINDOW)
                                             of Browser))
        else (FB.PROMPTWPRINT Browser T FileName " is not an
open NoteFile!!!"))

```

```

(NC.FileBrowserCheckpoint

```

[LAMBDA (Browser Key Item Menu)

(\* Randy.Gobbel " 4-Mar-87 16:13")

(\* \* Function called from file browser menu for notefile checkpoint.)

(\* \* rht 7/2/86%: Now passes InterestedWindow arg to NC.CheckpointDatabase.)

```
(for FileObject in (FB.SELECTEDFILES Browser) do (LET* ((FileName (MKATOM (FB.FETCHFILENAME FileObject)))
  (NoteFile (NC.NoteFileFromFileName FileName)))
  (if (type? NoteFile NoteFile)
    then (NC.CheckpointDatabase NoteFile NIL NIL
      (fetch (FILEBROWSER PROMPTWINDOW)
        of Browser))
    else (NC.PrintMsg NIL NIL FileName " is not an open
      NoteFile!!!" (CHARACTER 13]))
```

**(NC.FileBrowserAbort**

[LAMBDA (Browser Key Item Menu)

(\* rht%: " 2-Jul-86 19:05")

(\* \* Function called from file browser menu for notefile abort.)

(\* \* rht 7/2/86%: Now passes InterestedWindow arg to NC.AbortSession)

```
(for FileObject in (FB.SELECTEDFILES Browser) do (LET* ((FileName (MKATOM (FB.FETCHFILENAME FileObject)))
  (NoteFile (NC.NoteFileFromFileName FileName)))
  (if (type? NoteFile NoteFile)
    then (NC.AbortSession NoteFile (fetch (FILEBROWSER
      PROMPTWINDOW)
    of Browser))
    else (FB.PROMPTWPRINT Browser T FileName " is not an
      open NoteFile!!!" ])
```

**(NC.FileBrowserCompact**

[LAMBDA (Browser Key Item Menu InPlaceFlg)

(\* rht%: " 2-Jul-86 18:09")

(\* \* Function called from file browser menu for notefile compact.)

(\* \* rht 7/2/86%: Overhauled. Some code stolen from FB.COPY/RENAME.COMMAND.  
Much code stolen from FB.COPY/RENAME.MANY)

```
(LET
  ((FILELIST (FB.SELECTEDFILES Browser)))
  (COND
    ((NULL FILELIST))
    ((NULL (CDR FILELIST)) (* Just one file)
      (NC.CompactNoteFile (MKATOM (FB.FETCHFILENAME (CAR FILELIST)))
        NIL InPlaceFlg (fetch (FILEBROWSER PROMPTWINDOW) of Browser)))
    (T
      (* Many files to compact. Use code from FB.COPY/RENAME.MANY with call to FB.COPY/RENAME.ONE replaced by
        NC.CompactnoteFile.)

      (PROG (PREFIX FIELDS SUBDIR RETAIN)
        (COND
          ((NULL (SETQ PREFIX (FB.PROMPTFORINPUT (CONCAT "Compact " (LENGTH FILELIST)
            " files to which directory? ")
            (OR (fetch (FILEBROWSER DEFAULTDIR) of Browser)
              (DIRECTORYNAME T))
            Browser T))) (* Aborted)
          )
          ((STRPOS "*" PREFIX)
            (FB.PROMPTWPRINT Browser "Sorry, patterns not supported"))
          ([AND (OR (LISTGET (SETQ FIELDS (UNPACKFILENAME.STRING PREFIX))
            'HOST)
              (LISTGET FIELDS 'DIRECTORY)
              (LISTGET FIELDS 'DEVICE))
            (OR (LISTGET FIELDS 'NAME)
              (LISTGET FIELDS 'EXTENSION)
              (LISTGET FIELDS 'VERSION) (* Not a pure directory specification, and not just a simple
                directory name)
              (FB.PROMPTWPRINT Browser "Not a well-formed directory specification."))
            (T (replace (FILEBROWSER DEFAULTDIR) of Browser with (PACKFILENAME.STRING 'BODY FIELDS
              'DIRECTORY
              (DIRECTORYNAME T)))

            (* Now make sure the files are sorted by increasing version, so that multiple versions get copied in the right order)

            (SETQ SUBDIR (fetch (FBFILEDATA SUBDIRECTORY) of (fetch TIDATA of (CAR FILELIST)
              [COND
                ((for ITEM in (CDR FILELIST) thereis (NOT (STRING-EQUAL SUBDIR
                  (fetch (FBFILEDATA SUBDIRECTORY)
                    of (fetch TIDATA of ITEM)
                  (FB.PROMPTWPRINT Browser "Selected files are in multiple subdirectories")
                  (SETQ RETAIN (FB.PROMPTFORINPUT
                    (CONCAT "Retain subdirectory names below level of "
                      [for ITEM in (CDR FILELIST)
```

```

do [SETQ SUBDIR (FB.GREATEST.PREFIX SUBDIR
                    (fetch (FBFILEDATA SUBDIRECTORY)
                        of (fetch TIDATA of ITEM])
                    finally (RETURN (OR SUBDIR (SETQ SUBDIR
                                                    (SUBSTRING (fetch (FILEBROWSER
                                                                    PATTERN)
                                                                    of Browser)
                                                                    1
                                                                    (fetch (FILEBROWSER NAMESTART
                                                                    )
                                                                    of Browser])
                                                                    "?")
                    "Yes" Browser T T))
(SETQ RETAIN (COND
              ((NULL RETAIN) (* Aborted)
               (RETURN))
              ((OR (STRING-EQUAL RETAIN "YES")
                   (STRING-EQUAL RETAIN "Y"))
               (SETQ SUBDIR (ADD1 (NCHARS SUBDIR)))
               (* First character that changes)
               T)
              ((OR (STRING-EQUAL RETAIN "NO")
                   (STRING-EQUAL RETAIN "N"))
               NIL)
              (T (FB.PROMPTWPRINT Browser "?? ...Aborted.")
                 (RETURN)
                 (SELECTQ (fetch (FILEBROWSER SORTBY) of Browser)
                          (FB.NAMES.INCREASING.VERSION (* Okay)
                           )
                          (FB.NAMES.DECREASING.VERSION
                           (SETQ FILELIST (FB.SORT.VERSIONS FILELIST (FUNCTION FB.INCREASING.VERSION))))
                          (SORT FILELIST (FUNCTION FB.NAMES.INCREASING.VERSION)))
                 (SETQ PREFIX (\ADD.CONNECTED.DIR PREFIX))
                 (for ITEM in FILELIST
                  do (LET ((OLDNAME (FB.FETCHFILENAME ITEM))
                          (NC.CompactNoteFile (MKATOM OLDNAME)
                                                  (MKATOM (PACKFILENAME.STRING 'DIRECTORY PREFIX 'DIRECTORY
                                                                [COND
                                                                (RETAIN
                                                                (* Subdirectory of name between common prefix and root)
                                                                (SUBSTRING OLDNAME SUBDIR
                                                                (SUB1 (fetch (FBFILEDATA STARTOFNAME)
                                                                of (fetch TIDATA of ITEM])
                                                                'VERSION NIL 'BODY OLDNAME)))
                                                                InPlaceFlg
                                                                (fetch (FILEBROWSER PROMPTWINDOW) of Browser]))

```

**(NC.FileBrowserInspect&Repair**

```
[LAMBDA (Browser Key Item Menu ReadSubstancesFlg) (* rht%: "17-Jul-86 11:50")
```

```
(* * Function called from file browser menu for notefile inspect&repair.)
```

```
(* * rht 7/1/86%: Now passes InterestedWindow arg to NC.ScavengerPhase1.)
```

```
(* * rht 7/17/86%: Now calls NC.InspectAndRepairNoteFile instead of NC.ScavengerPhase1.)
```

```
(for FileObject in (FB.SELECTEDFILES Browser) do (NC.InspectAndRepairNoteFile (MKATOM (FB.FETCHFILENAME
                                                                                      FileObject))
                                         ReadSubstancesFlg
                                         (fetch (FILEBROWSER PROMPTWINDOW) of Browser]))
```

**(NC.FileBrowserExpunge**

```
[LAMBDA (FBROWSER KEY ITEM MENU CMD) (* rht%: "2-Jul-86 18:20")
```

```
(* * Code stolen from FB.EXPUNGECOMMAND. Replaced call to DELFILE with call to NC.DeleteDatabaseFile.)
```

```
(PROG ((TBROWSER (fetch (FILEBROWSER TABLEBROWSER) of FBROWSER))
      (NDELETED 0)
      (FILES FILENAME FAILED FILE)
      (COND
       [(SETQ FILES (TB.COLLECT.ITEMS TBROWSER 'DELETED))
        (FB.PROMPTWPRINT FBROWSER T "Expunging deleted files...")
        [for ITEM in FILES do (COND
                              ((NC.DeleteDatabaseFile (SETQ FILENAME (FB.FETCHFILENAME ITEM))
                                                         (fetch (FILEBROWSER PROMPTWINDOW) of FBROWSER)
                                                         T)
                               (add NDELETED 1)
                               (FB.REMOVE.FILE TBROWSER FBROWSER ITEM)
                               (FB.UPDATE.COUNTERS FBROWSER 'BOTH))
                              (T (FB.PROMPTWPRINT FBROWSER T "Couldn't expunge " FILENAME)
                                 (SETQ FAILED T]
        (FB.PROMPTWPRINT FBROWSER (COND
                                ((EQ NDELETED 0)
                                 "%
No")
```

```

(T (CONCAT (COND
  (FAILED "%
  (T "done, ")
  NDELETED)))
  " files expunged.")
(COND
  (FAILED (COND
    (CMD (FB.PROMPTWPRINT FBROWSER " " CMD " aborted.)))
    (RETURN)
  (T (FB.PROMPTWPRINT FBROWSER T "No files were marked for deletion"))
  (RETURN T])

```

**(NC.GetFileBrowserPattern**

[LAMBDA (MainWindow OldPattern)

(\* rht%: " 2-Jul-86 22:34")

(\* \* Ask user to provide a new file browser pattern. Cancel if user provides an extension.  
Else, add .notefile extension.)

```

(LET* [(Pattern (NC.AskUser "New pattern? " NIL OldPattern T MainWindow T T))
  (PatternPlusExtension (AND Pattern (PACKFILENAME 'BODY Pattern 'EXTENSION 'NOTEFILE 'VERSION
    ' *
    ' NAME
    ' *])
  (PROG1 (if (EQ (U-CASE (FILENAMEFIELD PatternPlusExtension 'EXTENSION))
    'NOTEFILE)
    then (SETQ NC.FileBrowserPatterns (UNION NC.FileBrowserPatterns (LIST PatternPlusExtension)
      ))
    else (NC.PrintMsg MainWindow T "Pattern must have either" (CHARACTER 13)
      ".NOTEFILE extension or no extension.")
      (DISMISS 1000)
      NIL)
    (NC.ClearMsg MainWindow T])

```

**(NC.FileBrowserRecomputeWithNewPattern**

[LAMBDA (BROWSER)

(\* rht%: " 2-Jul-86 15:04")

(\* \* Code basically stolen from FB.NEWPATTERNCOMMAND and FB.GET.NEWPATTERN.  
Just changed to call NC.GetFileBrowserPattern.)

```

(LET (PATTERN)
  (COND
    ([AND (FB.MAYBE.EXPUNGE BROWSER "New Pattern")
      (SETQ PATTERN (LET* ((OLDPATTERN (fetch (FILEBROWSER PATTERN) of BROWSER))
        (NEWPATTERN (NC.GetFileBrowserPattern (fetch (FILEBROWSER PROMPTWINDOW)
          of BROWSER)
          OLDPATTERN)))
      (COND
        (NEWPATTERN (DIRECTORY.FILL.PATTERN NEWPATTERN])
        (FB.SETNEWPATTERN BROWSER PATTERN)
        (FB.UPDATEBROWSERITEMS BROWSER])

```

**(NC.FindFileBrowserWinForPattern**

[LAMBDA (Pattern NoteCardsIconWindow)

(\* Randy.Gobbel " 5-Nov-86 16:14")

(\* \* Look for a notefile file browser having given pattern.)

```

(bind FileBrowser for Win in (WINDOWPROP NoteCardsIconWindow 'FileBrowserWins)
  when [AND (WINDOWP Win)
    (OR (OPENWP Win)
      (OPENWP (WINDOWPROP Win 'ICONWINDOW])
    (SETQ FileBrowser (WINDOWPROP (OR (WINDOWPROP Win 'ICONFOR)
      Win)
      'FILEBROWSER))
    (EQ (U-CASE Pattern)
      (U-CASE (MKATOM (fetch (FILEBROWSER PATTERN) of FileBrowser]
  do (RETURN Win])

```

)

(DEFINEQ

**(NC.FlashFileBrowserWin**

[LAMBDA (Pattern NoteCardsIconWindow)

(\* rht%: " 2-Jul-86 16:07")

(\* \* Return the file browser window if any, corresponding to given pattern.  
Expand if shrunken and flash.)

(\* \* rht 5/2/86%: Now will reexpand shrunken notefile browser but won't recover window if it's been closed.  
Closing window screws up filebrowser.)

(\* \* rht 7/2/86%: Changed to call NC.FindFileBrowserWinForPattern.)

```
(LET ((FileBrowserWin (NC.FindFileBrowserWinForPattern Pattern NoteCardsIconWindow))
      IconWin)
  (if [AND FileBrowserWin (OPENWP (SETQ IconWin (WINDOWPROP FileBrowserWin 'ICONWINDOW]
      then (EXPANDW IconWin))
  (if (AND FileBrowserWin (OPENWP FileBrowserWin))
      then (FLASHW FileBrowserWin)
      FileBrowserWin
  else NIL])
)

(PUTPROPS NCINTERFACE FILETYPE :FAKE-COMPILE-FILE)

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

## FUNCTION INDEX

NC.AskLinkLabel .....	12	NC.MakeMenusReadOnly .....	21
NC.AskNoteCardType .....	13	NC.MakeMenusReadWrite .....	22
NC.AttachNoteFileName .....	47	NC.MakeNewStyleNoteCardsIcon .....	37
NC.AttachPromptWindow .....	10	NC.MakeNoteCardsIcon .....	29
NC.AttachPromptWindowOffScreen .....	10	NC.MakeNoteFileIconOperationsMenus .....	17
NC.BinLoopProcess .....	26	NC.MakeNoteFileNameIndicator .....	48
NC.BringUpNoteCardsIcon .....	29	NC.MakeOldStyleNoteCardsIcon .....	38
NC.CardImageObjP .....	27	NC.MakeTEditReadOnly .....	20
NC.CheckpointSession .....	31	NC.MakeTEditReadWrite .....	21
NC.ChooseTopLevelCard .....	15	NC.MaxPromptWindowWidth .....	12
NC.CloseNoteCards .....	39	NC.MoveStructure .....	41
NC.CloseSession .....	31	NC.NoteCardsIconAfterMoveFn .....	29
NC.CloseStructure .....	40	NC.NoteCardsIconButtonEventFn .....	34
NC.CopyButtonEventFn .....	26	NC.NoteCardsIconCursorEventFn .....	37
NC.CopyMenu .....	20	NC.NoteCardsIconFn .....	37
NC.CopyStructure .....	39	NC.NoteCardsIconWindowP .....	28
NC.CreateNoteFileMenuItems .....	18	NC.NoteCardsWindowP .....	28
NC.DeleteStructure .....	40	NC.NoteFileIconButtonEventFn .....	43
NC.DisplayFileCapacity .....	46	NC.NoteFileIconMiddleButtonFn .....	45
NC.DoNoteFileOp .....	35	NC.NoteFileIndicatorWinHeight .....	48
NC.ExpandBars .....	14	NC.NoteFileMenuWindowP .....	41
NC.FileBrowserAbort .....	50	NC.NoteFileOperations .....	45
NC.FileBrowserCheckpoint .....	49	NC.NoteFileTitleLeftWhenSelectedFn .....	16
NC.FileBrowserClose .....	49	NC.PopUpNoteFileName .....	45
NC.FileBrowserCompact .....	50	NC.PrintReadOnlyMsg .....	19
NC.FileBrowserExpunge .....	51	NC.PromptWindowOpenP .....	12
NC.FileBrowserInspect&Repair .....	51	NC.ReadOnlyTEditSelFn .....	20
NC.FileBrowserMenu .....	29	NC.RedisplayNoteCardsIconWindow .....	34
NC.FileBrowserOpen .....	49	NC.RemoveSketchMenuItems .....	15
NC.FileBrowserRecomputeWithNewPattern .....	52	NC.ResetCachedMenus .....	18
NC.FindFileBrowserWinForPattern .....	52	NC.ResetNoteCardsIcon .....	37
NC.FlashFileBrowserWin .....	52	NC.ResetNoteFileInterface .....	43
NC.GetFileBrowserPattern .....	52	NC.SelectionMenusWhenSelectedFn .....	15
NC.GetNoteFileMenu .....	19	NC.SelectNoteCards .....	22
NC.GETPROMPTWINDOW .....	10	NC.SelectNoteCardsCopyInsertFn .....	26
NC.InstallCopyButtonEventFn .....	16	NC.SetNoteFileMenu .....	19
NC.InstallTitleBarButtonEventFn .....	16	NC.SetupNoteFileInterface .....	41
NC.InstallTitleBarLeftMenu .....	16	NC.ShowNoteFiles .....	47
NC.ListOfNoteFilesMenu .....	30	NC.ShrukenIconButtonEventFn .....	27
NC.LogoutAdvice .....	9	NC.SortCardTypes .....	14
NC.MakeCardImageObject .....	27	NC.ZapBinLoopProcess .....	26
NC.MakeMainMenu .....	7	NoteCards .....	7

## VARIABLE INDEX

NC.BoxesIconBitmap .....	28	NC.LinkDisplayModesMenu .....	6
NC.CachedMenus .....	12	NC.LinkLabelMenu .....	6
NC.CardImageObjFns .....	22	NC.LocalInsertLinkMenuItem .....	32
NC.CardOpsItems .....	32	NC.MaxFileNameChars .....	7
NC.CardOpsMenu .....	34	NC.MenuTitleFont .....	7
NC.ClosedNoteFileMenu .....	7	NC.NewCardsIconBitmap .....	28
NC.DatabaseFileNameSuggestion .....	7	NC.NoteCardsIconBitmap .....	28
NC.DefaultLeftButtonMenuItems .....	32	NC.NoteCardsIconMask .....	28
NC.DefaultMiddleButtonMenuItems .....	31	NC.NoteCardsIconMenuBitMap .....	34
NC.DefaultNoteFileIconMiddleButtonItems .....	47	NC.NoteCardsIconPosition .....	7
NC.DesiredTypeOrdering .....	12	NC.NoteCardsIconShrunkenBitmap .....	28
NC.EditPropListMenu .....	32	NC.NoteCardsIconShrunkenMask .....	28
NC.FileBrowserDefaultPatterns .....	49	NC.NoteCardsIconWindow .....	28
NC.FileBrowserExpungeMenuItem .....	49	NC.NoteFileIconCloseOperations .....	31
NC.FileBrowserMenuItems .....	49	NC.NoteFileIconOpenOperations .....	31
NC.FileBrowserMenuItemsToAdd .....	48	NC.NoteFileIconOperationsMenuItems .....	31
NC.FileBrowserMenuItemsToKeep .....	48	NC.NoteFileOpsItems .....	33
NC.FileBrowserMenuItemsToRemove .....	48	NC.NoteFileOpsMenu .....	34
NC.FileBrowserPatterns .....	49	NC.NoticedNoteFilesMenu .....	12
NC.FileBrowserRecomputeMenuItem .....	49	NC.NoticedNoteFilesMenuNewItem .....	12
NC.FileNameFont .....	7	NC.OldStyleIconMenuBitmap .....	34
NC.FullFileNameFont .....	7	NC.OpenedNoteFileMenu .....	7
NC.GlobalInsertLinkMenuItem .....	32	NC.OtherOpsItems .....	33
NC.InitialCardOpsItems .....	34	NC.OtherOpsMenu .....	34
NC.InitialNoteFileIconCloseOperations .....	31	NC.PromptWindowMonitorLock .....	7
NC.InitialNoteFileIconOpenOperations .....	31	NC.SelectionInProgress .....	22
NC.InitialNoteFileIconOperationsMenuItems .....	31	NC.ShowPropListMenu .....	32
NC.InitialNoteFileOpsItems .....	34	NC.TEditMenus .....	7
NC.InitialOtherOpsItems .....	34		

---

PROPERTY INDEX

NCINTERFACE .....53

---