

File created: 5-Nov-2020 19:32:56 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCGRAPHCARD.;5

previous date: 9-Jan-94 20:00:29 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCGRAPHCARD.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1985, 1986, 1987, 1988, 1989, 1990, 1993, 1994, 2020 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **NCGRAPHCARDCOMS**

```
([DECLARE%: DONTEVAL@LOAD FIRST (P (NC.LoadFileFromDirectories 'GRAPHER 'LISPUSERSDIRECTORIES)
(NC.LoadFileFromDirectories 'GRAPHERPATCH])
```

;;; Graph Card Menu Items

```
(GLOBALVARS NC.GCGraphEditMenuItems NC.GCGraphEditUnfixedMenuItems)
[INITVARS [NC.GCGraphEditMenuItems (NC.ExpandBars '(("Move Node" NC.BrowserMoveNodeFn "Moves a graph
node." (SUBITEMS ("Move Single Node"
NC.BrowserMoveNodeFn
"Move a graph node.")
("Move Node & SubTree"
NC.BrowserMoveSubtreeFn
"Move a subtree of nodes
following the movement
of the root.")
("Move Region"
NC.BrowserMoveNodesInRegionFn
"Move all nodes within a
specified region to
another region.)))
("Remove Node" NC.BrowserRemoveNodeFn "Remove a
graph node.")
("Connect Nodes" NC.BrowserAddEdgeFn "Connect two
graph nodes.")
("Disconnect Nodes" NC.BrowserRemoveEdgeFn
"Disconnect two graph nodes.")
BAR
("Add Label" NC.BrowserAddLabelFn "Add a string
label to graph.")
("Change Label" NC.BrowserChangeLabelFn "Change a
label.")
("Smaller Label" NC.BrowserShrinkLabelFn "Reduce
font size for a label.")
("Larger Label" NC.BrowserGrowLabelFn "Increase font
size for a label.")
BAR
("Toggle Shade" NC.BrowserToggleShadeFn "Inverts
label shade.")
("Toggle Border" NC.GCToggleBoxFn "Inverts label
box.")
BAR
("Directed/Undirected" NC.GCDirected/UndirectedFn)
("Sides/Centers" NC.GCSides/CentersFn)
BAR
(FIX% MENU NC.GCFixGraphEditMenuFn "Attach this menu
to edge of graph window.])
(NC.GCGraphEditUnfixedMenuItems (NC.ExpandBars '(("Move Node" NC.BrowserMoveNodeFn "Moves a graph
node." (SUBITEMS ("Move Single Node"
NC.BrowserMoveNodeFn
"Move a graph node.")
("Move Node & SubTree"
NC.BrowserMoveSubtreeFn
"Move a subtree of nodes
following the movement of
the root.")
("Move Region"
NC.BrowserMoveNodesInRegionFn
"Move all nodes
within a specified
region to another
region.)))
("Remove Node" NC.BrowserRemoveNodeFn "Remove a
graph node.")
("Connect Nodes" NC.BrowserAddEdgeFn "Connect
two graph nodes.")
("Disconnect Nodes" NC.BrowserRemoveEdgeFn
"Disconnect two graph nodes.")
BAR
("Add Label" NC.BrowserAddLabelFn "Add a string
```

```

label to graph.")
("Change Label" NC.BrowserChangeLabelFn
 "Change a label.")
("Smaller Label" NC.BrowserShrinkLabelFn
 "Reduce font size for a label.")
("Larger Label" NC.BrowserGrowLabelFn
 "Increase font size for a label.")
BAR
("Toggle Shade" NC.BrowserToggleShadeFn
 "Inverts label shade.")
("Toggle Border" NC.GCToggleBoxFn "Inverts label
 box.")
BAR
("Directed/Undirected"
 NC.GCDirected/UndirectedFn)
("Sides/Centers" NC.GCSides/CentersFn]

```

```

(FNS NCAddStub.GraphCard)
(GLOBALVARS NC.GraphCardIcon NC.ShrunkenGraphIconBM NC.ShrunkenGraphIconMask
 NC.ShrunkenGraphIconTemplate NC.UseDeletedLinkIconIndicatorsFlg TEDIT.ICON.FONT)

```

;;; Basic functions.

```

(FNS NC.MakeGraphCard NC.BringUpGraphCard NC.RelayGraphCard NC.CollectReferencesInGraph
 NC.GraphCardCloseFn NC.GraphCardLeftButtonFn NC.GraphCardMiddleButtonFn NC.GraphCopySubstance
 NC.MarkGraphDirty NC.GraphDirtyP NC.GraphNodeLinkIconUpdate NC.GetGraphSubstance
 NC.PutGraphSubstance NC.UpdateLinkImagesInGraph NC.DelReferencesToCardFromGraph
 NC.GraphChangeLabelFn NC.MakeGraphCardReadOnly NC.MakeGraphCardReadWrite NC.SaveOrRevertGraphCard)

```

;;; Hooks into Grapher for NoteCards applications

```

(FNS NC.GRAPHERCOPYBUTTONEVENTFN NC.INSIDEGRAPHERNODEP NC.GraphAddNodeFn NC.GraphDeleteNodeFn
 NC.GraphAddLinkFn NC.GraphDeleteLinkFn NC.GraphMoveNodeFn NC.GraphFontChangeFn NC.GraphAddNode
 NC.GraphRemoveNode NC.GraphAddLabel NC.InsertEdgeInGraph NC.InsertLabelInGraph
 NC.TrackCursorInGrapherWindow)
(FNS NC.GraphNodeIDGetProp NC.GraphNodeIDPutProp NC.GraphNodeIDRemProp)

```

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

```

(FNS NC.TranslateWindowPositionToGraphPosition NC.InsertLinkInGraph)

```

;;; New Graph Card Menu Fns

```

(FNS NC.GraphEditMenuWhenSelectedFn NC.GraphCardEditEventFn NC.GraphCardEditCommandFn
 NC.GCSetUpGraphEditMenus NC.GCFixGraphEditMenuFn NC.GCToggleBoxFn NC.GCDirected/UndirectedFn
 NC.GCSides/CentersFn)

```

;;; Other Graph Card Menu Fns moved from BROWSERCARD since both use many of these

```

(FNS NC.CoerceToGraphNodeID NC.CardFromBrowserNodeID NC.CoerceToGraphNodeIDOrLabel
 NC.SmashGraphNodeIDProps NC.SelectGraphNode NC.DisplayGraphLinksBetween)
(FNS NC.SetUpGraphEditMenus NC.GetGraphEditMenu NC.BrowserRightButtonFn NC.BrowserCreateCardFn
 NC.BrowserAddLabelFn NC.BrowserChangeLabelFn NC.BrowserAddNodeFn NC.BrowserCreateLinkFn
 NC.BrowserCreateGlobalLinkFn NC.BrowserAddLink NC.BrowserAddGlobalLink NC.BrowserAddEdgeFn
 NC.BrowserDeleteCardFn NC.BrowserRemoveNodeFn NC.BrowserDeleteLinkFn NC.BrowserRemoveEdgeFn
 NC.BrowserShrinkLabelFn NC.BrowserGrowLabelFn NC.BrowserMoveNodeFn NC.BrowserToggleShadeFn
 NC.CursorInsideGraphNodeP NC.BrowserMoveNodesInRegionFn NC.BrowserMoveSubtreeFn
 NC.BrowserFixGraphEditMenuFn NC.BrowserCreateCard NC.BrowserCreateLink NC.BrowserDeleteLink
 NC.BrowserAddNode NC.BrowserAddLabel NC.BrowserAddEdge NC.BrowserRemoveNode
 NC.DelBrowserContentsLink NC.BrowserRemoveEdge NC.BrowserChangeLabel NC.BrowserGetConfirmation)

```

;;; Stuff for shrinking graphs and browsers.

```

(BITMAPS NC.GraphIconBM NC.GraphIconMaskBM NC.GraphCardIcon)
(FNS NC.GraphCardShrinkFn)

```

;;; Stuff for making copies of graphs for making imageobj's.

```

(FNS NC.MakeExternalGraphCopy)

```

;;; Add the Graph card to the cardtypes list.

```

(FNS NC.AddGraphCard)
(DECLARE%: DONTEVAL@LOAD (P (NC.AddGraphCard)))
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
 NCGRAPHCARD))

```

```

(DECLARE%: DONTEVAL@LOAD FIRST

```

```

(NC.LoadFileFromDirectories 'GRAPHER 'LISPUSERSDIRECTORIES)

```

(NC.LoadFileFromDirectories 'GRAPHERPATCH)  
)

;;; Graph Card Menu Items

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.GCGraphEditMenuItems NC.GCGraphEditUnfixedMenuItems)  
)

(RPAQ? **NC.GCGraphEditMenuItems**

[NC.ExpandBars ' ("Move Node" NC.BrowserMoveNodeFn "Moves a graph node." (SUBITEMS ("Move Single Node"  
NC.BrowserMoveNodeFn  
"Move a graph node."  
("Move Node & SubTree"  
NC.BrowserMoveSubtreeFn  
"Move a subtree of nodes  
following the movement  
of the root."  
("Move Region"  
NC.BrowserMoveNodesInRegionFn  
"Move all nodes  
within a specified  
region to another  
region.")))  
  
("Remove Node" NC.BrowserRemoveNodeFn "Remove a graph node.")  
("Connect Nodes" NC.BrowserAddEdgeFn "Connect two graph nodes.")  
("Disconnect Nodes" NC.BrowserRemoveEdgeFn "Disconnect two graph nodes.")  
BAR  
("Add Label" NC.BrowserAddLabelFn "Add a string label to graph.")  
("Change Label" NC.BrowserChangeLabelFn "Change a label.")  
("Smaller Label" NC.BrowserShrinkLabelFn "Reduce font size for a label.")  
("Larger Label" NC.BrowserGrowLabelFn "Increase font size for a label.")  
BAR  
("Toggle Shade" NC.BrowserToggleShadeFn "Inverts label shade.")  
("Toggle Border" NC.GCToggleBoxFn "Inverts label box.")  
BAR  
("Directed/Undirected" NC.GCDirected/UndirectedFn)  
("Sides/Centers" NC.GCSides/CentersFn)  
BAR  
(FIX% MENU NC.GCFixGraphEditMenuFn "Attach this menu to edge of graph window.")]

(RPAQ? **NC.GCGraphEditUnfixedMenuItems**

[NC.ExpandBars ' ("Move Node" NC.BrowserMoveNodeFn "Moves a graph node." (SUBITEMS ("Move Single Node"  
NC.BrowserMoveNodeFn  
"Move a graph node."  
("Move Node & SubTree"  
NC.BrowserMoveSubtreeFn  
"Move a subtree of nodes  
following the movement  
of the root."  
("Move Region"  
NC.BrowserMoveNodesInRegionFn  
"Move all nodes  
within a specified  
region to another  
region.")))  
  
("Remove Node" NC.BrowserRemoveNodeFn "Remove a graph node.")  
("Connect Nodes" NC.BrowserAddEdgeFn "Connect two graph nodes.")  
("Disconnect Nodes" NC.BrowserRemoveEdgeFn "Disconnect two graph nodes.")  
BAR  
("Add Label" NC.BrowserAddLabelFn "Add a string label to graph.")  
("Change Label" NC.BrowserChangeLabelFn "Change a label.")  
("Smaller Label" NC.BrowserShrinkLabelFn "Reduce font size for a label.")  
("Larger Label" NC.BrowserGrowLabelFn "Increase font size for a label.")  
BAR  
("Toggle Shade" NC.BrowserToggleShadeFn "Inverts label shade.")  
("Toggle Border" NC.GCToggleBoxFn "Inverts label box.")  
BAR  
("Directed/Undirected" NC.GCDirected/UndirectedFn)  
("Sides/Centers" NC.GCSides/CentersFn)]

(DEFINEQ

(**NCAddStub.GraphCard**

[LAMBDA NIL

; Edited 3-Dec-87 18:58 by rht:

(\* \* kirk 18Jun86 Add the Graph card stub)

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

(**DECLARE** (GLOBALVARS NC.GraphCardIcon))

(NC.AddCardTypeStub 'Graph 'NoteCard 'NCGRAPHCARD NIL '(DisplayedInMenuFlg T)  
(LinkIconAttachedBitMap ,NC.GraphCardIcon))

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.GraphCardIcon NC.ShrunkenGraphIconBM NC.ShrunkenGraphIconMask NC.ShrunkenGraphIconTemplate
NC.UseDeletedLinkIconIndicatorsFlg TEDIT.ICON.FONT)
)

::: Basic functions.

(DEFINEQ

**(NC.MakeGraphCard**

[LAMBDA (Card Title NoDisplayFlg ParamList InterestedWindow RegionOrPosition) ; Edited 13-Jan-89 14:55 by sye

:: Creates a StructEdit card. A lot like a graph card except that it allows editing of underlying NC structure, and doesn't allow an 'insertlink'
operation in title bar menu.
:: fgh 11/14/85 Updated to handle Card object.
:: rht 5/5/86: Replaced call to NC.SetupTitleBarMenu with Install calls.
:: fgh 6/30/86 Added NC.GRAPHERCOPYBUTTONEVENTFN to SHOWGRAPH call
:: rht 9/19/86: Mod to above fix. Now calls NC.MakeNewCardWindow a new fn that creates window and hangs Card off windowprop.
:: pmi 6/23/87: Added the GRAPH.CHANGELABELFN NC.GraphChangeLabelFn to the Graph being created, which will then mark the substance
dirty when a label is changed.
:: pmi 4/5/88: Now passes a NIL TOPJUSTIFYFLG to SHOWGRAPH. This puts the origin at the lower left, the same as in browsers.
:: rht 6/9/88: Added RegionOrPosition arg and passed to NC.MakeNewCardWindow. Also added ParamList and InterestedWindow args which we
ignore.
:: Sye Jan/13/89: Added RIGHTBUTTONFN "NC.GraphCardShrinkFn" after calling showgraph to leave Grapher's editing menu untouched

(LET (Window Graph CardType)
(SETQ Graph (create GRAPH
GRAPH.ADDNODEFN \_ (FUNCTION NC.GraphAddNodeFn)
GRAPH.DELETENODEFN \_ (FUNCTION NC.GraphDeleteNodeFn)
GRAPH.ADDLINKFN \_))
(NC.SetSubstance Card Graph)
(COND
((NOT NoDisplayFlg)
(SETQ Window (NC.MakeNewCardWindow Card (OR Title "Untitled")
NIL NIL RegionOrPosition))
(SETQ CardType (NC.RetrieveType Card))
(SHOWGRAPH Graph Window (FUNCTION NC.GraphCardLeftButtonFn)
(FUNCTION NC.GraphCardMiddleButtonFn)
NIL T (FUNCTION NC.GRAPHERCOPYBUTTONEVENTFN))
(WINDOWPROP Window 'RIGHTBUTTONFN (FUNCTION NC.GraphCardEditEventFn))
(WINDOWADDPROP Window 'SHRINKFN (FUNCTION NC.GraphCardShrinkFn))
(NC.InstallTitleBarButtonEventFn Window (FUNCTION NC.TitleBarButtonEventFn))
(NC.InstallTitleBarLeftMenu Window CardType)
(NC.InstallTitleBarMiddleMenu Window CardType)))
(OR Window Card]))

**(NC.BringUpGraphCard**

[LAMBDA (Card Substance RegionOrPosition) ; Edited 13-Jan-89 16:23 by sye

::: Given a graph (or browser) Substance, open a grapher (browser) window and set it up to be a NoteCard with ID.
:: fgh 11/14/85 Updated to handle Card object.
:: rht 5/5/86: Now calls NC.InstallTitleBarButtonEventFn, NC.InstallTitleBarLeftMenu, and NC.InstallTitleBarMiddleMenu instead of
NC.SetupTitleBarMenu.
:: fgh 6/30/86 Added NC.GRAPHERCOPYBUTTONEVENTFN to SHOWGRAPH call
:: rht 11/16/86: Now moves window to RegionOrPosition if already up.
:: dwm 1/15/88 added ReadOnliness
:: pmi 2/18/88: Removed one of the two FLASHWs. Fixed up check for same position to be a little more efficient.

(LET (Region Title GraphWindow IconWindow CardType)
(COND
((SETQ GraphWindow (NC.FetchWindow Card))
(AND (WINDOWP (WINDOWPROP GraphWindow 'ICONFOR))
(SETQ GraphWindow (EXPANDW GraphWindow)))
(TOTOPW GraphWindow)
[if RegionOrPosition
then (if (REGIONP RegionOrPosition)
then (if [NOT (EQUAL RegionOrPosition (WINDOWPROP GraphWindow 'REGION])
then (SHAPEW GraphWindow (NC.DetermineDisplayRegion Card RegionOrPosition))
elseif (POSITIONP RegionOrPosition)
then (SETQ WindowRegion (WINDOWPROP GraphWindow 'REGION))
(if [NOT (AND (EQUAL (fetch (POSITION XCOORD) of RegionOrPosition)
(fetch (REGION LEFT) of WindowRegion))
(EQUAL (fetch (POSITION YCOORD) of RegionOrPosition)
(fetch (REGION BOTTOM) of WindowRegion))
then (SHAPEW GraphWindow (NC.DetermineDisplayRegion Card RegionOrPosition])
(FLASHW GraphWindow))

```
(T (NC.SetUserDataProp Card 'ReadOnly (NC.CardReadOnlyOpenP Card))
  (SETQ GraphWindow (SHOWGRAPH Substance (CREATEW (NC.DetermineDisplayRegion Card RegionOrPosition)
                                                    (NC.RetrieveTitle Card))
                    (FUNCTION NC.GraphCardLeftButtonFn)
                    (FUNCTION NC.GraphCardMiddleButtonFn)
                    NIL T (FUNCTION NC.GRAPHERCOPYBUTTONEVENTFN)))
  (WINDOWPROP GraphWindow 'RIGHTBUTTONFN (FUNCTION NC.GraphCardEditEventFn))
  (SETQ CardType (NC.RetrieveType Card))
  (WINDOWADDPROP GraphWindow 'SHRINKFN (FUNCTION NC.GraphCardShrinkFn))
  (NC.InstallTitleBarButtonEventFn GraphWindow (FUNCTION NC.TitleBarButtonEventFn))
  (NC.InstallTitleBarLeftMenu GraphWindow CardType)
  (NC.InstallTitleBarMiddleMenu GraphWindow CardType)))
GraphWindow])
```

**(NC.RelayoutGraphCard**

[LAMBDA (Win) (\* pmi%: "21-Jul-87 18:42")

- (\* \* Relayout the graph by making roots out of all nodes without FROMNODES.)
- (\* \* rht 8/15/86%: Now uses proper COPYBUTTONEVENTFN in the SHOWGRAPH call.)
- (\* \* rht 11/1/86%: Added NC.ProtectedCardOperation wrapper and check for ops in progress.)
- (\* \* rg |3/4/87| rewritten for new version of NC.ProtectedCardOperation)
- (\* \* pmi 7/21/87%: Made sure LAYOUTGRAPH is getting a list for ROOTIDS.)

```
(LET ((Card (NC.CoerceToCard Win)))
  (NC.ProtectedCardOperation
   Card "Relayout Graph Card" NIL
   (LET ((Graph (WINDOWPROP Win 'GRAPH))
         (GraphNodes)
         (if (SETQ GraphNodes (fetch (GRAPH GRAPHNODES) of Graph))
             then (SETQ Graph (LAYOUTGRAPH GraphNodes
                                     [MKLIST (OR (for Node in GraphNodes
                                                when (NULL (fetch (GRAPHNODE FROMNODES) of Node))
                                                collect (fetch (GRAPHNODE NODEID) of Node))
                                                (fetch (GRAPHNODE NODEID) of (CAR GraphNodes)
                                                  'Lattice))
                                     (replace (GRAPH GRAPH.ADDNODEFN) of Graph with (FUNCTION NC.GraphAddNodeFn))
                                     (replace (GRAPH GRAPH.DELETENODEFN) of Graph with (FUNCTION NC.GraphDeleteNodeFn))
                                     (replace (GRAPH GRAPH.ADDLINKFN) of Graph with (FUNCTION NC.GraphAddLinkFn))
                                     (replace (GRAPH GRAPH.DELETELINKFN) of Graph with (FUNCTION NC.GraphDeleteLinkFn))
                                     (replace (GRAPH GRAPH.MOVENODEFN) of Graph with (FUNCTION NC.GraphMoveNodeFn))
                                     (replace (GRAPH GRAPH.FONTCHANGEFN) of Graph with (FUNCTION NC.GraphFontChangeFn))
                                     (SHOWGRAPH Graph Win (FUNCTION NC.GraphCardLeftButtonFn)
                                                  (FUNCTION NC.GraphCardMiddleButtonFn)
                                                  NIL T (FUNCTION NC.GRAPHERCOPYBUTTONEVENTFN))
                                     (NC.InstallTitleBarButtonEventFn Win (FUNCTION NC.TitleBarButtonEventFn))
```

**(NC.CollectReferencesInGraph**

[LAMBDA (Card CheckAndDeleteFlg ReturnLinkIconsFlg ReturnLocationsFlg) (\* rht%: "1-Jun-87 22:11")

- (\* \* Return a list of all links or link icons in graph substance Substance. If CheckAndDeleteFlg, then delete any links found that are not valid links.)
- (\* \* fgh |11/14/85| Updated to handle Card object.)
- (\* \* rht 5/26/87%: Changed to match reduced functionality of NC.ValidLinkP, now have to check that destination of ActualLink is a valid card.)

```
(DECLARE (GLOBALVARS NC.DeletedLinkImageObject))
(LET ((Substance (NC.FetchSubstance Card))
      ActualLink DirtyFlg Links LinkIcon CollectItem)
  [SETQ Links (for GraphNode in (fetch (GRAPH GRAPHNODES) of Substance)
              when (COND
                    ((NC.LinkIconImageObjP (SETQ LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode))
                                             )
                     (SETQ ActualLink (NC.FetchLinkFromLinkIcon LinkIcon))
                     (COND
                      ((NULL CheckAndDeleteFlg)
                       ((AND (LISTP CheckAndDeleteFlg)
                            (MEMB (fetch (Link DestinationCard) of ActualLink)
                                   CheckAndDeleteFlg)))
                      [(AND (NC.ValidLinkP ActualLink)
                           (NC.ValidCardP (fetch (Link DestinationCard) of ActualLink)
                                             (replace (GRAPHNODE NODELABEL) of GraphNode with NC.DeletedLinkImageObject)
                                             (SETQ DirtyFlg T)
                                             NIL)))
                      (T NIL))
                    collect (PROGN (SETQ CollectItem (COND
                                      (ReturnLinkIconsFlg LinkIcon)
                                      (T ActualLink)))
                                (COND
```

```

                ( (NULL ReturnLocationsFlg)
                  CollectItem)
                (T (CONS CollectItem (fetch (GRAPHNODE NODELABEL) of GraphNode]
(CONS Links DirtyFlg])

```

**(NC.GraphCardCloseFn**

[LAMBDA (CardIdentifier) (\* rht%: "16-Oct-86 16:56")

(\* Called when a grapher-based note card is being closed. Writes graph to database and does other clean up functions)

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

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

```

(LET (Window (Card (NC.CoerceToCard CardIdentifier)))
      (SETQ Window (NC.FetchWindow Card))
      (AND Window (for AttachedWindow in (ALLATTACHEDWINDOWS Window) do (DETACHWINDOW AttachedWindow)
              (CLOSEW AttachedWindow)))
      (AND Window (REMOVEPROMPTWINDOW Window])

```

**(NC.GraphCardLeftButtonFn**

[LAMBDA (GraphNode GraphWindow) ; Edited 27-Jun-88 12:25 by Trigg

;; If this is a node containing a link icon, then do: IF Selection in progress, then pressing a node gets its destination ID if its actually a note card node.  
 ;; Otherwise, bring up the note card pointed to by the node.

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

;; fgh 11/14/85 Isn't this obsolete???????

;; rht&pmi 11/14/86: No longer does stupid pushcopy stuff.

;; rg 11/4/87 added ReadOnly to NC.EditNoteCard call

;; jrc 13-jun-88 added the equivalent to RHTPATCH322 -- use the LinkIconLeftButtonFn of the card type, if there is one

;; rht&jrc 6/27/88: Replaced core code with call to NC.LinkIconLeftButtonFn.

```

(LET ((LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode)))
      (if [AND (NC.LinkIconImageObjP LinkIcon)
               (NOT (OR (KEYDOWNP 'LSHIFT)
                       (KEYDOWNP 'RSHIFT)
                       (KEYDOWNP 'CTRL)
                       (KEYDOWNP 'COPY)
                       (KEYDOWNP 'DELETE)
                       (KEYDOWNP 'MOVE]
                then (NC.LinkIconLeftButtonFn LinkIcon GraphWindow)
                else NIL])

```

**(NC.GraphCardMiddleButtonFn**

[LAMBDA (GraphNode GraphWindow) (\* Randy.Gobbel "14-Nov-86 16:42")

(\* IF Selection in progress, then pressing a node gets its destination ID if its actually a note card node. Otherwise, jump to normal button processing for an link icon)

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

(\* \* fgh |11/14/85| Isn't this obsolete?????????)

(\* \* rht&pmi 11/14/86%: No longer does stupid NC.SpecialBrowserMiddleButtonFn stuff.)

```

(LET ((LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode)))
      (if [AND (NC.LinkIconImageObjP LinkIcon)
               (NOT (OR (KEYDOWNP 'LSHIFT)
                       (KEYDOWNP 'RSHIFT)
                       (KEYDOWNP 'CTRL)
                       (KEYDOWNP 'COPY)
                       (KEYDOWNP 'DELETE)
                       (KEYDOWNP 'MOVE]
                then (NC.LinkIconMiddleButtonFn LinkIcon GraphWindow)
                else NIL])

```

**(NC.GraphCopySubstance**

[LAMBDA (Card FromStream ToStream Length) (\* rht%: "23-Jan-86 12:12")

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

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

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

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

(\* \* Copy the bytes)

```

(LET* ((FromStartPtr (GETFILEPTR FromStream))

```

```

    (FromEndPtr (PLUS Length FromStartPtr))
    (COPYBYTES FromStream ToStream FromStartPtr FromEndPtr)
T])

```

**(NC.MarkGraphDirty**

```

[LAMBDA (Card ResetFlg) (* fgh%: "14-Nov-85 21:25")

(* * Mark or unmark graph as having been changed -- Sorry no can do!!!)

NIL])

```

**(NC.GraphDirtyP**

```

[LAMBDA (Graph) (* rht%: "1-Feb-85 15:39")

(* * A graph is always considered to be dirty since we can't tell whether its changed or not.)

(* * rht 2/1/85%: For now we can't tell by looking at graph substance whether we're dirty or not.
Leave it up to the SubstanceDirtyFlg on the card.)

NIL])

```

**(NC.GraphNodeLinkIconUpdate**

```

[LAMBDA (Window GraphNode) (* rht%: "29-Oct-87 11:59")

(* * Check whether the image box size agrees with graphnode size.
If not, update graph node. REDISPLAYW should eventually happen to the graph after calling this.)

(* * rht&pmi 2/6/87%: Now includes BorderSize in calculation of graph node label size.
This is important for virtual nodes.)

(* * rht 10/29/87%: Now handles pointer icons as well as link icons.)

(LET ((LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode))
[BorderSize (TIMES 2 (GRAPHNODE/BORDER/WIDTH (fetch (GRAPHNODE NODEBORDER) of GraphNode]
(GraphNodeWidth (fetch (GRAPHNODE NODEWIDTH) of GraphNode))
(GraphNodeHeight (fetch (GRAPHNODE NODEHEIGHT) of GraphNode))
ImageBox XSize YSize)
(if [OR [AND (NC.LinkIconImageObjP LinkIcon)
(SETQ ImageBox (NC.LinkIconImageBoxFn LinkIcon (WINDOWPROP Window 'DSP]
(AND (NC.PointerIconImageObjP LinkIcon)
(SETQ ImageBox (NC.PointerIconImageBoxFn LinkIcon (WINDOWPROP Window 'DSP]
then (SETQ XSize (fetch (IMAGEBOX XSIZE) of ImageBox))
(SETQ YSize (fetch (IMAGEBOX YSIZE) of ImageBox))
(if [AND GraphNodeWidth (NOT (EQUAL XSize (DIFFERENCE GraphNodeWidth BorderSize]
then (replace (GRAPHNODE NODEWIDTH) of GraphNode with (PLUS XSize BorderSize))
(if [AND GraphNodeHeight (NOT (EQUAL YSize (DIFFERENCE GraphNodeHeight BorderSize]
then (replace (GRAPHNODE NODEHEIGHT) of GraphNode with (PLUS YSize BorderSize])

```

**(NC.GetGraphSubstance**

```

[LAMBDA (Card Length Stream VersionNum) (* fgh%: "25-May-86 18:04")

(* * Read the Graph)

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

(* * fgh |11/20/85| NoteCards now passes start and end ptrs down.)

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

(* * rht 1/23/86%: Now takes Stream as arg instead of computing from Card.)

(* * rht 2/7/86%: Now uses READGRAPH instead of HREAD)

(* * rht 2/28/86%: Added special handling for old version -1 style.
I.e. pre 1.3k.)

(* * fgh |5/1/86| Can't have negative version numbers, so changed old style versions to be 255.0)

(* * fgh |5/25/86| Undid preceeding change. Turns out -1 never came from file only passed as Arg from converter.)

(if (EQP VersionNum -1)
then (HREAD Stream)
else (READGRAPH Stream])

```

**(NC.PutGraphSubstance**

```

[LAMBDA (Card Stream) (* rht%: "14-Feb-86 18:09")
(* Put Graph in card ID onto DatabaseStream)

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

(* * fgh |11/20/85| NoteCards now takes care of setting the start and end pointers for the substance.)

```

(\* rht 1/23/86%: Now takes Stream as arg instead of computing from Card.)  
(\* fgh |2/6/86| Now returns a version number)  
(\* rht 2/7/86%: Now uses DUMPGRAPH rather than HPRINT)

(LET ((Graph (NC.FetchSubstance Card))) (\* Clean up BITMAPS in Graph data structure)  
(for GraphNode in (fetch GRAPHNODES of Graph) do (replace (GRAPHNODE NODELABELBITMAP) of GraphNode  
with NIL) (\* Write data structure)  
(DUMPGRAPH Graph Stream)  
0))

**(NC.UpdateLinkImagesInGraph**

[LAMBDA (SourceCard DestinationCard) (\* rht%: "29-Oct-87 12:02")

(\* Update all the NoteCard links that point to DestinationID from the graph card specified by SourceCardID)  
(\* rht 2/15/85%: Added hack to redraw links into and out of the affected node.  
This is in case it has changed size. Note the calls to internal grapher functions!)  
(\* rht 10/3/85%: Threw away above mentioned code. It was screwing up the link-params info for the affected nodes.)  
(\* fgh |11/17/85| Updated to handle card object.)  
(\* rht 10/29/87%: Now handles pointer icons as well as link icons.)

(LET (Window Graph GraphNodes LinkIcon OldAddLinkFn OldDeleteLinkFn)  
(SETQ Window (NC.FetchWindow SourceCard))  
(SETQ Graph (NC.FetchSubstance SourceCard))  
(for GraphNode in (SETQ GraphNodes (fetch (GRAPH GRAPHNODES) of Graph))  
when [OR [AND (NC.LinkIconImageObjP (SETQ LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode))  
(NC.SameCardP DestinationCard (fetch (Link DestinationCard) of (NC.FetchLinkFromLinkIcon  
LinkIcon)  
(AND (NC.PointerIconImageObjP LinkIcon)  
(NC.SameCardP DestinationCard (fetch (NC.Pointer DestinationCard) of (NC.FetchPointerFromPointerIcon  
LinkIcon])  
do (\* Update the size of the graph node if doesn't agree with image  
box.)  
(NC.GraphNodeLinkIconUpdate Window GraphNode))  
(REDISPLAYW Window])

**(NC.DelReferencesToCardFromGraph**

[LAMBDA (SourceCard LinkOrDestinationCard Don'tCreateDeletedImageObjFlg) (\* rht%: "4-Nov-86 20:58")

(\* Delete from the graph specified by Graph all link icon nodes whose DESTINATIONID is eq to DestinationID)  
(\* rht 11/17/84%: Now only redisplays graph if ID is both active \*and\* has a window.)  
(\* kirk 14Nov85%: deleted use of LinkID)  
(\* rht 4/30/86%: Now uses value of NC.UseDeletedLinkIconIndicators to determine whether to delete node or replace with Deleted indicator.)  
(\* rht 11/4/86%: Now takes Don'tCreateDeletedImageObjFlg arg.)

(DECLARE (GLOBALVARS NC.UseDeletedLinkIconIndicatorsFlg NC.DeletedLinkImageObject))  
(LET ((ImageBox (NC.DeletedLinkImageBoxFn NC.DeletedLinkImageObject))  
LinkIcon  
(LinkFlg (type? Link LinkOrDestinationCard))  
DestinationCard Graph)  
(COND  
[LinkFlg (OR (NC.CardP SourceCard)  
(SETQ SourceCard (fetch (Link SourceCard) of LinkOrDestinationCard)  
(T (SETQ DestinationCard LinkOrDestinationCard)))]  
(SETQ Graph (NC.FetchSubstance SourceCard))  
(for GraphNode in (fetch (GRAPH GRAPHNODES) of Graph)  
when [AND (NC.LinkIconImageObjP (SETQ LinkIcon (fetch (GRAPHNODE NODELABEL) of GraphNode))  
(COND  
(LinkFlg (NC.SameLinkP LinkOrDestinationCard (NC.FetchLinkFromLinkIcon LinkIcon))  
(T (NC.SameCardP (fetch (Link DestinationCard) of (NC.FetchLinkFromLinkIcon LinkIcon)  
DestinationCard])  
do (if (AND NC.UseDeletedLinkIconIndicatorsFlg (NOT Don'tCreateDeletedImageObjFlg))  
then (replace (GRAPHNODE NODELABEL) of GraphNode with NC.DeletedLinkImageObject)  
(replace (GRAPHNODE NODEWIDTH) of GraphNode with (fetch (IMAGEBOX XSIZE) of ImageBox))  
(replace (GRAPHNODE NODEHEIGHT) of GraphNode with (fetch (IMAGEBOX YSIZE) of ImageBox))  
else (NC.GraphRemoveNode GraphNode Graph (NC.FetchWindow SourceCard)  
T)))]  
(COND  
((AND (NC.ActiveCardP SourceCard)  
(NC.FetchWindow SourceCard))  
(REDISPLAYGRAPH (NC.FetchWindow SourceCard))



**(NC.GraphChangeLabelFn**

[LAMBDA (Graph Window Node)

(\* pmi%: "23-Jun-87 17:32")

(\* pmi 6/23/87%: Created to mark the card as dirty and prompt for the node label - needed to make sure substance is marked dirty when a label is changed.)

```
(LET ((GraphID (NC.CoerceToCard Window)))
      (NC.MarkCardDirty GraphID)
      (PROMPTINWINDOW "Node label? "])
```

**(NC.MakeGraphCardReadOnly**

[LAMBDA (Card)

; Edited 28-Jan-88 07:50 by MacDonald

(\* \* Make a GraphCard Read-Only.)

```
(DECLARE (GLOBALVARS NC.ShowPropListMenu))
(LET ((Window (NC.FetchWindow Card))
      PropListEditor)
      (NC.ProtectedCardOperation Card "Make Read-Only" Window (if (NC.CardSomehowDirtyP Card)
                                                                    then (NC.SaveOrRevertGraphCard Card Window))
      (NC.SetUserDataProp Card 'ReadOnly T)
      (if (WINDOWP (SETQ PropListEditor (NC.PropListEditorOpenP Window)))
          then (NC.MakeTEditReadOnly PropListEditor)
              (WINDOWPROP PropListEditor 'TEDIT.MENU NC.ShowPropListMenu))
      (WINDOWPROP Window 'RIGHTBUTTONFN NIL)
      (NC.MakeMenusReadOnly Window '(Show% Links Show% Info Designate% FileBoxes |Close and Save|
                                     Relayout% Graph |Edit Property List|
                                     '(Indicate% NoteFile |Close w/o Saving| |Close and Save| |Save in NoteFile|))
```

**(NC.MakeGraphCardReadWrite**

[LAMBDA (Card)

; Edited 28-Jan-88 07:53 by MacDonald

(\* \* Make a GraphCard Read-Write)

```
(DECLARE (GLOBALVARS NC.EditPropListMenu))
(LET ((Window (NC.FetchWindow Card))
      PropListEditor)
      (NC.ProtectedCardOperation Card "Make Read-Write" Window (if (WINDOWP (SETQ PropListEditor (
                                                                                               NC.PropListEditorOpenP
                                                                                               Window)))
                                                                    then (NC.MakeTEditReadWrite PropListEditor
                                                                    )
                                                                    (WINDOWPROP PropListEditor
                                                                    'TEDIT.MENU
                                                                    NC.EditPropListMenu))
      (WINDOWPROP Window 'RIGHTBUTTONFN (FUNCTION GRAPHEDITEVENTFN))
      (NC.MakeMenusReadWrite Window)
      (NC.SetUserDataProp Card 'ReadOnly NIL])
```

**(NC.SaveOrRevertGraphCard**

[LAMBDA (Card Window)

; Edited 27-Jan-88 08:04 by MacDonald

```
(ALLOW.BUTTON.EVENTS)
(if (MENU (create MENU
              TITLE _ "Card has been edited"
              ITEMS _ '(("Save Changes" T "Save edited version of card in NoteFile")
                        ("Revert" NIL "Discard edits, refetch card contents from NoteFile")))
    then (NC.CardSaveFn Card NIL Window)
    else (NC.GetNoteCard Card)
          (SHOWGRAPH (NC.FetchSubstance Card)
                     Window
                     (FUNCTION NC.GraphCardLeftButtonFn)
                     (FUNCTION NC.GraphCardMiddleButtonFn)
                     NIL T (FUNCTION NC.GRAPHERCOPYBUTTONONEVENTFN))
          (NC.MarkCardDirty Card T])
```

;;; Hooks into Grapher for NoteCards applications

(DEFINEQ

**(NC.GRAPHERCOPYBUTTONONEVENTFN**

[LAMBDA (Window)

(\* rht%: "15-Jul-86 18:12")

(\* \* Handle a COPYBUTTONONEVENT in a NC Grapher window. If the button is down over a node and the TTY PROCESS is a selecting cards process, process it as such, otherwise do the standard grapher copy button event fn of copyinserting the whole graph.)

(\* \* fgh |6/30/86| First created.)

(\* rht 7/15/86%: Now calls NC.MakeExternalGraphCopy to smash NC links.)

```
(LET ((Graph (WINDOWPROP Window 'GRAPH))
      InNode PositionAndLastState TTYWindow (Position (CONS 0 0)))
  (if [NOT (AND [WINDOWP (SETQ TTYWindow (PROCESSPROP (TTY.PROCESS)
                                                       'WINDOW]
              (WINDOWPROP TTYWindow 'SelectingCards]
      then
        (* The current TTY PROCESS is not a selecting cards process --
        insert a copy of the graph with the nc links smashed.)

        (COPYINSERT (GRAPHEROBJ (NC.MakeExternalGraphCopy Graph)))

      else
        (* We want to handle this special for NoteCards to allow shift-select of link icons in the graph)

        (* First track the cursor until its let up)

        (SETQ PositionAndLastState (NC.TrackCursorInGrapherWindow Window Graph))

        (* Cursor up -- determine what to do)

        (if (NULL (CAR PositionAndLastState))
            then

          (* Let up outside the window -- get out of here)

          NIL

        else

          (* Find the node the mouse was when the buttons were let up --
          if any)

          (SETQ InNode (for GraphNode in (fetch (GRAPH GRAPHNODES) of Graph)
              thereis (NC.INSIDEGRAPHERNODEP (PROGN (replace (POSITION XCOORD) of Position
                                                           with (LASTMOUSEX Window))
                                                           (replace (POSITION YCOORD) of Position
                                                           with (LASTMOUSEY Window))
                                                           Position)
              GraphNode)))

          (if [OR (NULL InNode)
                (NULL (NC.LinkIconImageObjP (fetch (GRAPHNODE NODELABEL) of InNode))]
              then

                (* Let up inside the window, but not in a node. Or the Node is not a LinkIcon node.
                Do the standard COPYINSERTFN for GRAPHER)

                (COPYINSERT (GRAPHEROBJ (NC.MakeExternalGraphCopy Graph)))

              else

                (* Let up inside a node, must want to select it.)

                (COPYINSERT (fetch (GRAPHNODE NODELABEL) of InNode])
```

**(NC.INSIDEGRAPHERNODEP**

[LAMBDA (Position Node) (\* Randy.Gobbel "19-Nov-86 14:31")

(\* Is position inside of graphernode Node)  
 (\* fgh |6/30/86| First created.)  
 (\* rg |11/19/86| Bug fix%: Node was being referred to as GraphNode)

```
(LET ((Left (GN/LEFT Node))
      Bottom (GN/BOTTOM Node)))
  (AND (GEQ (fetch (POSITION XCOORD) of Position)
        Left)
       (LEQ (fetch (POSITION XCOORD) of Position)
            (PLUS Left (fetch (GRAPHNODE NODEWIDTH) of Node)))
       (GEQ (fetch (POSITION YCOORD) of Position)
        Bottom)
       (LEQ (fetch (POSITION YCOORD) of Position)
            (PLUS Bottom (fetch (GRAPHNODE NODEHEIGHT) of Node)))
       Node])
```

**(NC.GraphAddNodeFn**

[LAMBDA (Graph Window) (\* fgh%: "18-Nov-85 00:19")

(\* Add a node, normal or link, to a graph)  
 (\* rht 11/9/84%: Now checks result of NC.MakeLink before building link.)

```
(PROG (LinkLabel Link (GraphCard (NC.CoerceToCard Window)))
  (RETURN (COND
```

```

((NULL (SETQ Link (WINDOWPROP Window 'NoteCardInsertingLink NIL)))
(NC.MarkCardDirty GraphCard)
(DEFAULT.ADDNODEFN Graph Window))
([OR (type? Link Link)
(AND (SETQ LinkLabel (NC.AskLinkLabel Window NIL NIL T T))
(SETQ Link (NC.MakeLink Window LinkLabel))
(NC.MarkCardDirty GraphCard)
(NODECREATE (for bind NodeID unless (FASSOC (SETQ NodeID (GENSYM))
(fetch (GRAPH GRAPHNODES) of Graph))
do (RETURN NodeID))
(NC.MakeLinkIcon Link)
(CURSORPOSITION NIL Window))

```

**(NC.GraphDeleteNodeFn**

[LAMBDA (Node Graph Window) (\* rht%: "13-May-87 12:32")

```

(* * Just mark the card as dirty.)
(* * rht 4/30/86%: Now makes sure there's a window first.)
(* * rht 5/13/87%: Now deletes link if node is a link icon.)

```

```

(LET ((GraphNodeLabel (fetch (GRAPHNODE NODELABEL) of Node)))
(AND Window (NC.MarkCardDirty (NC.CoerceToCard Window))))
(if (NC.LinkIconImageObjP GraphNodeLabel)
then (NC.DeleteLink (NC.FetchLinkFromLinkIcon GraphNodeLabel)
NIL T))

```

**(NC.GraphAddLinkFn**

[LAMBDA (From To Graph Window) (\* rht%: "15-Nov-85 16:05")

(\* \* Just mark the card as dirty.)

```

(PROG ((GraphID (NC.CoerceToCard Window)))
(NC.MarkCardDirty GraphID))

```

**(NC.GraphDeleteLinkFn**

[LAMBDA (From To Graph Window) (\* rht%: "30-Apr-86 15:04")

```

(* * Just mark the card as dirty.)
(* * rht 4/30/86%: Now makes sure there's a window first.)

```

```

(AND Window (NC.MarkCardDirty (NC.CoerceToCard Window]))

```

**(NC.GraphMoveNodeFn**

[LAMBDA (Node Newpos Graph Window) (\* rht%: "15-Nov-85 16:05")

(\* \* Just mark the card as dirty.)

```

(PROG ((GraphID (NC.CoerceToCard Window)))
(NC.MarkCardDirty GraphID))

```

**(NC.GraphFontChangeFn**

[LAMBDA (How Node Graph Window) (\* rht%: "15-Nov-85 16:05")

(\* \* Just mark the card as dirty.)

```

(PROG ((GraphID (NC.CoerceToCard Window)))
(NC.MarkCardDirty GraphID))

```

**(NC.GraphAddNode**

[LAMBDA (Graph GraphWindow) ; Edited 12-Apr-88 17:26 by pmi

```

;; pmi 4/12/88: Called from Grapher to create a new node for Graph. This is the addnodefn used by NC.InsertLinkInGraph.
;; pmi 8/26/88: fixed typo where Window was used instead of GraphWindow.

```

```

(DECLARE (GLOBALVARS NC.MsgDelay))
(NC.ProtectedCardOperation (NC.CoerceToCard GraphWindow)
"Add Node" GraphWindow (NCP.WithLockedCards (PROG (NewNodeData SourceCard LinkOrLinkLabel
DestinationCard DisplayMode Position
GraphNodeID Card Node)

```

;; Get information to add a node from the graph's window.

```

(SETQ NewNodeData (WINDOWPROP GraphWindow
'NewNodeData))
(SETQ SourceCard (LISTGET NewNodeData 'SourceCard))
(SETQ LinkOrLinkLabel (LISTGET NewNodeData
'LinkOrLinkLabel))
(SETQ DestinationCard (LISTGET NewNodeData
'DestinationCard))
(SETQ DisplayMode (LISTGET NewNodeData

```

```

'DisplayMode))
(SETQ Position (LISTGET NewNodeData 'Position))
(if [SETQ Link
(COND
((type? Link LinkOrLinkLabel)
LinkOrLinkLabel)
(T (NC.MakeLink GraphWindow LinkOrLinkLabel
DestinationCard SourceCard
DisplayMode]
then ;; Check to see if the node has already been added to the
;; graph.
; Create hash array if haven't already.
(NC.GetBrowserHashArray SourceCard Graph)
[SETQ GraphNodeID
(NC.GetBrowserNodeID
SourceCard
(SETQ Card (fetch (Link DestinationCard)
of Link]
(COND
((for GraphNode
in (fetch (GRAPH GRAPHNODES)
of Graph)
thereis (EQ GraphNodeID (
NC.CoerceToGraphNodeID
GraphNode)))
(NC.PrintMsg GraphWindow T "Node for
card '" (NC.RetrieveTitle Card)
"' already in graph.")
(DISMISS NC.MsgDelay)
(NCP.DeleteLinks Link)
(RETURN NIL)))
(NC.MarkCardDirty SourceCard)
(SETQ Node
(NODECREATE GraphNodeID (NC.MakeLinkIcon
Link)
(create POSITION
XCOORD _ -500
YCOORD _ -500)))
(RETURN Node])

```

**(NC.GraphRemoveNode**

```

[LAMBDA (Node Graph Window QuietFlg) (* rht%: "30-Apr-86 16:19")
(* * Remove this node from the graph. This code taken from Grapher's EDITDELETENODE)
(* * rht 4/30/86%: Added QuietFlg and other minor fixes.)
(for ToNode in (TOLINKS Node) bind OtherNode do (* Undisplay links.)
(SETQ OtherNode (GETNODEFROMID
(if (EQ (CAR ToNode)
LINKPARAMS)
then (CADR ToNode)
else ToNode)
(fetch (GRAPH GRAPHNODES) of Graph)))
(AND Window (NC.DisplayGraphLinksBetween Node OtherNode Window
Graph))
(GRAPHDELETELINK Node OtherNode Graph Window))
(for FromNode in (FROMLINKS Node) bind OtherNode do (* Undisplay links.)
(SETQ OtherNode (GETNODEFROMID
(if (EQ (CAR FromNode)
LINKPARAMS)
then (CADR FromNode)
else FromNode)
(fetch (GRAPH GRAPHNODES) of Graph)))
(AND Window (NC.DisplayGraphLinksBetween OtherNode Node
Window Graph))
(GRAPHDELETELINK OtherNode Node Graph Window))
(AND Window (DISPLAYNODE Node (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Window Graph))
(GRAPHDELETENODE Node Graph Window)
(OR QuietFlg (NC.PrintMsg NIL T "Node " (fetch (GRAPHNODE NODELABEL) of Node)
" removed."))
Node])

```

**(NC.GraphAddLabel**

```

[LAMBDA (Graph GraphWindow) ; Edited 8-Apr-88 15:12 by pmi
;; This builds a label node in the default grapher style except that we force non-boxed nodes.
;; pmi 6/23/87: Added call to NC.MarkCardDirty to make sure substance is marked dirty when nodes are created.
;; pmi 4/8/88: Now called from Grapher to create a new label node for Graph. This is the addnodefn used by NC.InsertLabelInGraph.
(LET ((GraphCard (NCP.CardFromWindow GraphWindow)))
(NC.ProtectedCardOperation GraphCard "Add Label" GraphWindow

```

```
(NCP.WithLockedCards (LET (NewLabelData Label Position Font Border LabelShade GraphNodes
                           NodeName)
                          (SETQ NewLabelData (WINDOWPROP GraphWindow 'NewLabelData))
                          (SETQ Label (LISTGET NewLabelData 'Label))
                          (SETQ Position (LISTGET NewLabelData 'Position))
                          (SETQ Font (LISTGET NewLabelData 'Font))
                          (SETQ Border (LISTGET NewLabelData 'Border))
                          (SETQ LabelShade (LISTGET NewLabelData 'LabelShade))
                          (NC.MarkCardDirty (NC.CoerceToCard GraphWindow))
                          (SETQ GraphNodes (fetch (GRAPH GRAPHNODES) of Graph))
                          (while (FASSOC (SETQ NodeName (PACK* Label (GENSYM)))
                                         GraphNodes))
                           (NODECREATE NodeName Label (create POSITION
                                                             XCOORD _ -500
                                                             YCOORD _ -500)
                           NIL NIL (OR Font (CHOOSE.GRAPH.FONT Graph))
                           Border LabelShade])
```

**(NC.InsertEdgeInGraph**

```
[LAMBDA (GraphCard FromNode ToNode) ; Edited 8-Apr-88 15:25 by pmi
;; Adds a new edge to GraphCard, using FromNode as the source and ToNode as the destination. Will only allow edge addition if there are
;; currently no edges between the two nodes.
(LET* [(Window (NCP.CardWindow GraphCard))
       (Graph (WINDOWPROP Window 'GRAPH)
              (NC.GetBrowserHashArray GraphCard Graph))
       (COND
        ((EQ FromNode ToNode)
         (NC.PrintMsg NIL T "Can't add edge from a node to itself.")
         (FLASHW PROMPTWINDOW))
        ((OR (MEMBTONODES (fetch (GRAPHNODE NODEID) of ToNode)
                    (fetch (GRAPHNODE TONODES) of FromNode))
         (MEMBTONODES (fetch (GRAPHNODE NODEID) of FromNode)
                    (fetch (GRAPHNODE TONODES) of ToNode)))
         (NC.PrintMsg NIL T "There is already an edge between those two nodes.")
         (FLASHW PROMPTWINDOW))
        (T (NC.MarkCardDirty GraphCard)
           (ADD/AND/DISPLAY/LINK FromNode ToNode Window Graph)))
       NIL])
```

**(NC.InsertLabelInGraph**

```
[LAMBDA (GraphCard Label Position Font Border LabelShade) ; Edited 5-Apr-88 17:55 by pmi
;; Insert a Label in GraphWindow at Position
;; Called from NCP.LocalGlobalLink to add a graph label to an existing card at a specified position. Just sets up the proper ADDNODEFN in graph
;; record, attaches arguments to the graph window, and invokes the grapher editor.
;; pmi 8/26/88: Restores the original GRAPH.ADDNODEFN to this Graph after inserting this label.
(LET* [(Window (NCP.CardWindow GraphCard))
       (Graph (WINDOWPROP Window 'GRAPH)
              (WINDOWPROP Window 'NewLabelData (LIST 'Label Label 'Position Position 'Font Font 'Border Border
              'LabelShade LabelShade))
       (RESETLST
        [RESETSAVE NIL `(PROGN (replace (GRAPH GRAPH.ADDNODEFN) of Graph with (FUNCTION ,(fetch (GRAPH
                                                                                               GRAPH.ADDNODEFN
                                                                                               )
                                                                                               of Graph]
        (replace (GRAPH GRAPH.ADDNODEFN) of Graph with (FUNCTION NC.GraphAddLabel))
        (EDITADDNODE Window Position))])
```

**(NC.TrackCursorInGrapherWindow**

```
[LAMBDA (Window Graph) (* fgh%: "30-Jun-86 13:44")
(* * Follow the cursor around a graph window inverting nodes that its in until the mouse buttons are let up.)
(* * fgh |6/30/86| First created.)
(LET [Buttons NewInNode InNode (Position (CONS 0 0))
      (DS (WINDOWPROP Window 'DSP)
      [until (MOUSESTATE UP) do
(* * Track the cursor until the mouse is up)
(SETQ Buttons LASTMOUSEBUTTONS)
(SETQ InNode NewInNode)
(* * Figure out if in a node)
(SETQ NewInNode (for GraphNode in (fetch (GRAPH GRAPHNODES) of Graph)
                thereis (NC.INSIDEGRAPHNODEP (PROGN (replace (POSITION XCOORD)
                                                            of Position
                                                            with (LASTMOUSEX
                                                            Window))
                (replace (POSITION YCOORD)
```

of Position  
with (LASTMOUSEY  
Window))  
Position))

```

GraphNode)))

(* * Uninvert the old node, invert the new node.)
      (if (NEQ InNode NewInNode)
          then (AND InNode (FLIPNODE InNode DS))
              (AND NewInNode (FLIPNODE NewInNode DS])

(* * Uninvert anything that needs it)
(AND NewInNode (FLIPNODE NewInNode DS))

(* * Return a record of the mouse location at button up and the last mouse button that were down before let up)
(CONS (CONS (LASTMOUSEX Window)
            (LASTMOUSEY Window))
      Buttons])
)

```

(DEFINEQ

**(NC.GraphNodeIDGetProp**

[LAMBDA (GraphNodeID PropName)

; Edited 6-Dec-88 12:31 by krivacic

::: Do a GETPROP either to the UID or to the atom if it's a label.

```

(if (type? UID GraphNodeID)
    then (NC.UIDGetProp GraphNodeID PropName)
    else (GETPROP GraphNodeID PropName])

```

**(NC.GraphNodeIDPutProp**

[LAMBDA (GraphNodeID PropName Value)

; Edited 28-Nov-88 13:53 by krivacic

(\* \* Do a PUTPROP either to the UID or to the atom if it's a label.)

```

(if (type? UID GraphNodeID)
    then (NC.UIDPutProp GraphNodeID PropName Value)
    else (PUTPROP GraphNodeID PropName Value])

```

**(NC.GraphNodeIDRemProp**

[LAMBDA (GraphNodeID PropName)

(\* rht%: " 6-Jul-86 17:23")

(\* \* Do a REMPROP either to the UID or to the atom if it's a label.)

```

(if (type? UID GraphNodeID)
    then (NC.UIDRemProp GraphNodeID PropName)
    else (REMPROP GraphNodeID PropName])
)

```

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

(DEFINEQ

**(NC.TranslateWindowPositionToGraphPosition**

[LAMBDA (Card Window WindowPositionX WindowPositionY)

(\* fgh%: " 6-Feb-86 22:15")

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

```

(create POSITION
  XCOORD _ WindowPositionX
  YCOORD _ WindowPositionY])

```

**(NC.InsertLinkInGraph**

[LAMBDA (GraphCard LinkOrLinkLabel DestinationCard DisplayMode Position)

; Edited 8-Apr-88 14:47 by pmi

:: Insert a Link to DestinationCard in GraphCard at Position

:: pmi 4/8/88: Called from NCP.LocalGlobalLink to add a graph node to an existing card at a specified position. Just sets up the proper  
:: ADDNODEFN in graph record, attaches arguments to the graph window, and invokes the grapher editor.

:: pmi 8/26/88: Restores the original GRAPH.ADDNODEFN to this Graph after inserting this link.

```

(LET* [(Window (NCP.CardWindow GraphCard))
      (Graph (WINDOWPROP Window 'GRAPH)
             (WINDOWPROP Window 'NewNodeData (LIST 'SourceCard GraphCard 'LinkOrLinkLabel LinkOrLinkLabel
                                                    'DestinationCard DestinationCard 'DisplayMode DisplayMode
                                                    'Position Position))])

```

(RESETLST

```
[RESETSAVE NIL `(PROGN (replace (GRAPH GRAPH.ADDNODEFN) of Graph with (FUNCTION , (fetch (GRAPH
GRAPH.ADDNODEFN
)
)
of Graph]
(replace (GRAPH GRAPH.ADDNODEFN) of Graph with (FUNCTION NC.GraphAddNode)
(EDITADDNODE Window Position)))]
```

)

::: New Graph Card Menu Fns

(DEFINEQ

(NC.GraphEditMenuWhenSelectedFn

[LAMBDA (MenuItem Menu MouseKey)

; Edited 11-Jan-89 16:46 by rtk

::: bk 1/11/89: Fix bug when the divider bars "----" item is selected.

::: Called when a selection is made in the graph editor menu. Just call the function, passing it the main window as arg.

```
(AND (CADR MenuItem)
(APPLY* (CADR MenuItem)
(GETMENUPROP Menu 'MAINWINDOW]))
```

(NC.GraphCardEditEventFn

[LAMBDA (GRWINDOW)

; Edited 13-Jan-89 10:58 by sye

(\* implements a graph editor on the right button transition of a window.)

```
(COND
((NOT (INSIDE? (DSPCLIPPINGREGION NIL GRWINDOW)
(LASTMOUSEX GRWINDOW)
(LASTMOUSEY GRWINDOW)))
(DOWINDOWCOM GRWINDOW))
((SHIFTDOWNP 'CTRL)
(TRACKNODE GRWINDOW))
((EQ (NC.GraphCardEditCommandFn GRWINDOW)
'STOP)
(CLOSEW GRWINDOW]))
```

(\* do menu)

(NC.GraphCardEditCommandFn

[LAMBDA (GRAPHWINDOW)

; Edited 13-Jan-89 12:50 by sye

```
(NC.GCSetUpGraphEditMenus GRAPHWINDOW)
(LET ((GCGraphEditUnfixableMenu (WINDOWPROP GRAPHWINDOW 'GCGRAPHEDITUNFIXABLEMENU))
(GCGraphEditFixableMenu (WINDOWPROP GRAPHWINDOW 'GCGRAPHEDITFIXABLEMENU))
(Card (NC.CoerceToCard GRAPHWINDOW))
(GCGraphEditFixedMenuWin)
(if (WINDOWPROP GRAPHWINDOW 'FIXED-GRAPH-MENU-WINDOW)
then (if (NC.FetchUserDataProp Card 'ReadOnly)
then (REMOVEWINDOW GCGraphEditFixedMenuWin)
(WINDOWPROP GRAPHWINDOW 'RIGHTBUTTONFN NIL))
else (PUTMENUPROP GCGraphEditUnfixableMenu 'MAINWINDOW GRAPHWINDOW)
(MENU GCGraphEditUnfixableMenu))
else (if (NC.FetchUserDataProp Card 'ReadOnly)
then (WINDOWPROP GRAPHWINDOW 'RIGHTBUTTONFN NIL)
else (PUTMENUPROP GCGraphEditFixableMenu 'MAINWINDOW GRAPHWINDOW)
(MENU GCGraphEditFixableMenu))
```

(NC.GCSetUpGraphEditMenus

[LAMBDA (Window)

; Edited 13-Jan-89 12:46 by sye

::: Create the "fixed" and "unfixed" menus for graph cards and set them up on the window unless they already exist.

(DECLARE (GLOBALVARS NC.MenuFont NC.GCGraphEditMenuItems NC.GCGraphEditUnfixedMenuItems))

; Edited 13-May-88 21:48 by Trigg

```
(LET [(GCGraphEditUnfixableMenu (WINDOWPROP Window 'GCGRAPHEDITUNFIXABLEMENU))
(GCGraphEditFixableMenu (WINDOWPROP Window 'GCGRAPHEDITFIXABLEMENU))
[OR GCGraphEditUnfixableMenu (WINDOWPROP Window 'GCGRAPHEDITUNFIXABLEMENU)
(SETQ GCGraphEditUnfixableMenu
(create MENU
ITEMS _ NC.GCGraphEditUnfixedMenuItems
TITLE _ "Editor Menu"
WHENSELECTEDFN _ (FUNCTION NC.GraphEditMenuWhenSelectedFn)
CHANGEOFFSETFLG _ T
CENTERFLG _ T
MENUOFFSET _ (CONS -1 0)
MENUFONT _ NC.MenuFont
ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
1)]
(OR GCGraphEditFixableMenu (WINDOWPROP Window 'GCGRAPHEDITFIXABLEMENU)
(SETQ GCGraphEditFixableMenu (create MENU using
```

GCGraphEditUnfixableMenu
ITEMS \_
NC.GCGraphEditMenuItems

])

**(NC.GCFixGraphEditMenuFn**

[LAMBDA (Window) ; Edited 13-Jan-89 11:52 by sye

```

;; Attach the Graph edit menu to the side of the graph window.
;; rht 9/20/85: Now uses ADDMENU offscreen so it doesn't flash in lower left corner before coming up.
;; rht 3/3/86: Now does some 'random' WINDOWPROPS to make reshaping browser not screw up the attached menu.
;; rht 3/7/86: Uses ATTACHMENU to attach the menu.
;; pmi 3/25/87: Added NC.MenuFont to all menus
;; rht 1/14/88: Now copies existing menu hung off window rather than building new one from scratch.
;; bk 1/11/89: Add FIXED-GRAPH-MENU-WINDOW window prop so links menu knows to re-display this menu.
(DECLARE (GLOBALVARS NC.MenuFont))
(LET ((ATTACHED-WINDOWS (WINDOWPROP Window 'ATTACHEDWINDOWS))
      GraphEditMenu GraphEditMenuWin)
  (SETQ GraphEditMenu (create MENU using (WINDOWPROP Window 'GCGRAPHEDITUNFIXABLEMENU)
                                       MENU COLUMNS _ 1))
  (PUTMENUPROP GraphEditMenu 'MAINWINDOW Window)
  (SETQ GraphEditMenuWin (ATTACHMENU GraphEditMenu Window 'RIGHT 'TOP))
  ; Rig so that close of menu won't close browser.
  (WINDOWDELPROP GraphEditMenuWin 'PASSTOMAINCOMS 'CLOSEW)
  (WINDOWPROP GraphEditMenuWin 'GRAPHEDITMENUFLG T)
  (WINDOWPROP Window 'FIXED-GRAPH-MENU-WINDOW T)
  (WINDOWPROP GraphEditMenuWin 'BROWSER-MAINWINDOW Window)
  (WINDOWADDPROP GraphEditMenuWin 'CLOSEFN [FUNCTION (LAMBDA (W)
                                                       (WINDOWPROP (WINDOWPROP W 'BROWSER-MAINWINDOW)
                                                                'FIXED-GRAPH-MENU-WINDOW NIL)
                                                                (DETACHWINDOW W)
                                                                T))])
)

```

**(NC.GCToggleBoxFn**

[LAMBDA (Window) ; Edited 13-Jan-89 13:24 by sye

```

;; Check if the menu is attached to the Links menu
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
          (if (fetch (GRAPH GRAPHNODES) of Graph)
              then (AND (EDITTOGGLEBORDER Window)
                        (NC.MarkCardDirty (NC.CoerceToCard Window)))
              else (FLASHW PROMPTWINDOW)
                  (NC.PrintMsg NIL T "Can't toggle border: graph has no nodes."))]
)

```

**(NC.GCDirected/UndirectedFn**

[LAMBDA (Window) ; Edited 13-Jan-89 13:29 by sye

```

;; Check if the menu is attached to the Links menu
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
          (if (fetch (GRAPH GRAPHNODES) of Graph)
              then (TOGGLE/DIRECTEDFLG Window)
                  (NC.MarkCardDirty (NC.CoerceToCard Window)))
              else (FLASHW PROMPTWINDOW)
                  (NC.PrintMsg NIL T "Graph has no nodes."))]
)

```

**(NC.GCSides/CentersFn**

[LAMBDA (Window) ; Edited 13-Jan-89 13:28 by sye

```

;; Check if the menu is attached to the Links menu
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
          (if (fetch (GRAPH GRAPHNODES) of Graph)
              then (TOGGLE/SIDESFLG Window)
                  (NC.MarkCardDirty (NC.CoerceToCard Window)))
              else (FLASHW PROMPTWINDOW)
                  (NC.PrintMsg NIL T "Graph has no nodes."))]
)

```

)

;;; Other Graph Card Menu Fns moved from BROWSECARD since both use many of these

(DEFINEQ

**(NC.CoerceToGraphNodeID**

[LAMBDA (NodeOrNodeID) (\* rht%: "26-Nov-85 01:25")

(\* \* Extract the graph node ID from a graph node, a virtual node, or a Link params thingie.)



```
(if (AND (LISTP NodeOrNodeID)
        (EQ (CAR NodeOrNodeID)
            LINKPARAMS))
    then (SETQ NodeOrNodeID (CADR NodeOrNodeID)))
(if (LISTP NodeOrNodeID)
    then (SETQ NodeOrNodeID (CAR NodeOrNodeID)))
(if (LISTP NodeOrNodeID)
    then (SETQ NodeOrNodeID (CAR NodeOrNodeID)))
(AND (type? UID NodeOrNodeID)
     NodeOrNodeID])
```

**(NC.CardFromBrowserNodeID**

[LAMBDA (BrowserNodeID) (\* rht%: " 6-Jul-86 17:27")

(\* \* Extract the card from a browser nodeID.)  
 (\* \* rht 6/10/86%: Now checks first for valid UID.)

```
(AND (type? UID BrowserNodeID)
     (NC.GraphNodeIDGetProp BrowserNodeID 'CardObject])
```

**(NC.CoerceToGraphNodeIDOrLabel**

[LAMBDA (GraphNode) (\* rht%: "20-Sep-85 12:54")

(\* \* This is like NC.CoerceToGraphNodeID except that it returns the label if the node is a label node.  
 Coerces virtual label node ids to labels.)

```
(if (NC.CoerceToGraphNodeID GraphNode)
    else (LET ((NodeID (fetch (GRAPHNODE NODEID) of GraphNode)))
            (if (LISTP NodeID)
                then (CAR NodeID)
                else NodeID]))
```

**(NC.SmashGraphNodeIDProps**

[LAMBDA (GraphNodeID) (\* rht%: " 5-Jun-87 15:34")

(\* \* Smash the prop list of GraphNodeID except be sure to save the CardObject prop if there is one.)  
 (\* \* rht 6/5/87%: Now checks to make sure GraphNodeID is a UID before trying to clear its proplist.)

```
(LET ((Card (NC.CardFromBrowserNodeID GraphNodeID)))
    (AND (type? UID GraphNodeID)
         (NC.UIDSetPropList GraphNodeID NIL))
    (AND Card (NC.GraphNodeIDPutProp GraphNodeID 'CardObject Card]))
```

**(NC.SelectGraphNode**

[LAMBDA (Window Graph Mess) ; Edited 22-Jan-88 18:04 by Trigg

;; User selects a node from the browser. The node is inverted and returned.  
 ;; rht 1/19/88: Now prints message to window and also checks to make sure that cursor is inside region of node before pronouncing it a valid  
 ;; selection.

```
(DECLARE (GLOBALVARS PROMPTWINDOW))
(RESETFORM (TTYDISPLAYSTREAM PROMPTWINDOW)
 (CLRSPROMPT)
 (PROG (NODE)
 (COND
 ((NOT (fetch (GRAPH GRAPHNODES) of Graph))
 (PROMPTPRINT " No nodes in graph.")
 (RETURN)))
 (NC.PrintMsg Window T Mess)
 (SETQ NODE (READ/NODE (fetch (GRAPH GRAPHNODES) of Graph)
                        Window))
 (NC.ClearMsg Window)
 (if (NC.CursorInsideGraphNodeP NODE Window)
     then (FLIPNODE NODE Window)
         (RETURN NODE)
     else (RETURN NIL]))
```

**(NC.DisplayGraphLinksBetween**

[LAMBDA (FromNode ToNode Window Graph) (\* rht%: "29-Apr-85 00:19")

(\* \* Display (or undisplay) the grapher links joining FromNode and ToNode.)

```
(PROG (LinkParams ReverseLinkParams)
 (COND
 ((SETQ LinkParams (LINKPARAMETERS FromNode ToNode))
 (DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION
                                             XCOORD _ 0
                                             YCOORD _ 0))
              Window Graph 1 LinkParams))
 ((SETQ ReverseLinkParams (LINKPARAMETERS ToNode FromNode))
 (DISPLAYLINK ToNode FromNode (CONSTANT (create POSITION
```

XCOORD \_ 0  
YCOORD \_ 0)

Window Graph 1 ReverseLinkParams])

)

(DEFINEQ

**(NC.SetUpGraphEditMenus**

```
[LAMBDA (Window) ; Edited 13-May-88 21:52 by Trigg
;; Create the "fixed" and "unfixed" menus for browser cards and set them up on the window unless they already exist.
(DECLARE (GLOBALVARS NC.MenuFont NC.GraphEditMenuItems NC.GraphEditUnfixedMenuItems) ; Edited 13-May-88 21:48 by Trigg
(LET [(GraphEditUnfixableMenu (WINDOWPROP Window 'GRAPHEDITUNFIXABLEMENU))
(GraphEditFixableMenu (WINDOWPROP Window 'GRAPHEDITFIXABLEMENU)
[OR GraphEditUnfixableMenu (WINDOWPROP Window 'GRAPHEDITUNFIXABLEMENU
(SETQ GraphEditUnfixableMenu
(create MENU
ITEMS _ NC.GraphEditMenuItems
TITLE _ "Editor Menu"
WHENSELECTEDFN _ (FUNCTION NC.GraphEditMenuWhenSelectedFn)
CHANGEOFFSETFLG _ T
CENTERFLG _ T
MENUOFFSET _ (CONS -1 0)
MENUFONT _ NC.MenuFont
ITEMHEIGHT _ (IPLUS (FONTPROP NC.MenuFont 'HEIGHT)
1]
(OR GraphEditFixableMenu (WINDOWPROP Window 'GRAPHEDITFIXABLEMENU
(SETQ GraphEditFixableMenu (create MENU using GraphEditUnfixableMenu
ITEMS _
NC.GraphEditUnfixedMenuItems
])
```

**(NC.GetGraphEditMenu**

```
[LAMBDA (Window) ; Edited 13-May-88 21:55 by Trigg
;; Create, if necessary, and bring up the graph editor menu.
;; pmi 3/25/87: Added NC.MenuFont to all menus
;; rht 1/14/88: Now ensures that both menus are computed rather than just the one needed.
;; dwm 1/27/88: Added ReadOnly stuff
;; rht 3/3/88: Fixed bug in dwm fix. Card var was unbound.
;; rht 5/13/88: Now calls NC.SetUpGraphEditMenus to build menus and stash on window rather than doing it in line.
```

```
(NC.SetUpGraphEditMenus Window)
(LET ((GraphEditUnfixableMenu (WINDOWPROP Window 'GRAPHEDITUNFIXABLEMENU))
(GraphEditFixableMenu (WINDOWPROP Window 'GRAPHEDITFIXABLEMENU))
(Card (NC.CoerceToCard Window))
(GraphEditUnfixableMenu GraphEditFixableMenu GraphEditFixedMenuWin)
;; If menu is already up, then put up a menu without the FIXMENU option, otherwise, allow user to fix menu.
(if [SETQ GraphEditFixedMenuWin (for Win in (ATTACHEDWINDOWS Window)
thereis (AND (OPENWP Win)
(WINDOWPROP Win 'GRAPHEDITMENUFLG)
then (if (NC.FetchUserDataProp Card 'ReadOnly)
then (REMOVEWINDOW GraphEditFixedMenuWin)
(WINDOWPROP Window 'RIGHTBUTTONFN NIL)
else (PUTMENUPROP GraphEditUnfixableMenu 'MAINWINDOW Window)
(MENU GraphEditUnfixableMenu))
else (if (NC.FetchUserDataProp Card 'ReadOnly)
then (WINDOWPROP Window 'RIGHTBUTTONFN NIL)
else (PUTMENUPROP GraphEditFixableMenu 'MAINWINDOW Window)
(MENU GraphEditFixableMenu))
```

**(NC.BrowserRightButtonFn**

```
[LAMBDA (Window) (* rht%: "17-Nov-85 21:18")
(* * Causes grapher editor menu to come up on right button. If menu is already fixed then brings up menu without FIXMENU item.)
(if (INSIDE? (DSPCLIPPINGREGION NIL Window)
(LASTMOUSEX Window)
(LASTMOUSEY Window))
then (NC.GetGraphEditMenu Window)
else (DOWINDOWCOM Window])
```

**(NC.BrowserCreateCardFn**

```
[LAMBDA (Window) ; Edited 9-Jan-89 15:46 by sye
;;; Called from graph edit menu to create card and node. Just sets up the proper ADDNODEFN in graph record and invokes grapher editor.
;; Check if the menu is attached to the Links menu
(LET (AttachW)
```

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(SETQ AttachW (NC.AttachPromptWindow Window NIL NIL NIL T))
(replace (GRAPH GRAPH.ADDNODEFN) of (WINDOWPROP Window 'GRAPH) with (FUNCTION NC.BrowserCreateCard))
(EDITADDDNODE Window NIL AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(CLOSEW AttachW])
```

**(NC.BrowserAddLabelFn**

[LAMBDA (Window)

; Edited 13-Jan-89 13:41 by sye

;;; Called from graph edit menu to add label to graph. Just sets up the proper ADDNODEFN in graph record and invokes grapher editor.

;; Check if the menu is attached to the Links menu

```
(LET (AttachW)
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(SETQ AttachW (NC.AttachPromptWindow Window))
(replace (GRAPH GRAPH.ADDNODEFN) of (WINDOWPROP Window 'GRAPH) with (FUNCTION NC.BrowserAddLabel))
(EDITADDDNODE Window NIL AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(CLOSEW AttachW])
```

**(NC.BrowserChangeLabelFn**

[LAMBDA (WINDOW)

; Edited 9-Jan-89 14:46 by rtk

;; Check if the menu is attached to the Links menu

```
(LET (AttachW)
(SETQ WINDOW (OR (WINDOWPROP WINDOW 'MAINWINDOW)
WINDOW))
(SETQ AttachW (NC.AttachPromptWindow WINDOW))
(replace (GRAPH GRAPH.CHANGELABELFN) of (WINDOWPROP WINDOW 'GRAPH) with (FUNCTION NC.BrowserChangeLabel))
(AND (EDITCHANGE LABEL WINDOW AttachW)
(NC.MarkCardDirty (NC.CoerceToCard WINDOW)))
(CLOSEW AttachW])
```

**(NC.BrowserAddNodeFn**

[LAMBDA (Window)

; Edited 15-Dec-88 15:59 by krivacic

;;; Called from graph edit menu to create browser node for an existing card. Just sets up the proper ADDNODEFN in graph record and invokes grapher editor.

;; Check if the menu is attached to the Links menu

```
(LET (AttachW)
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(SETQ AttachW (NC.AttachPromptWindow Window NIL NIL NIL T))
(replace (GRAPH GRAPH.ADDNODEFN) of (WINDOWPROP Window 'GRAPH) with (FUNCTION NC.BrowserAddNode))
(EDITADDDNODE Window NIL AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(CLOSEW AttachW])
```

**(NC.BrowserCreateLinkFn**

[LAMBDA (Window)

; Edited 15-Dec-88 16:00 by krivacic

;;; Called from graph edit menu to create link and edge.

;;; rht 4/20/87: No longer calls EDITAPPLYTOLINK if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET ((Graph (WINDOWPROP Window 'GRAPH))
AttachW)
(if (fetch (GRAPH GRAPHNODES) of Graph)
then (SETQ AttachW (NC.AttachPromptWindow Window))
(EDITAPPLYTOLINK (FUNCTION NC.BrowserAddLink)
'added
(WINDOWPROP Window 'GRAPH)
Window AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(CLOSEW AttachW)
else (FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "Can't create links: graph has no nodes."))
```

**(NC.BrowserCreateGlobalLinkFn**

[LAMBDA (Window)

; Edited 15-Dec-88 16:00 by krivacic

;;; Called from graph edit menu to create a global link and edge.

;;; rht 4/20/87: No longer calls EDITAPPLYTOLINK if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
```

```

(if (fetch (GRAPH GRAPHNODES) of Graph)
  then (EDITAPPLYTOLINK (FUNCTION NC.BrowserAddGlobalLink)
    'added
    (WINDOWPROP Window 'GRAPH)
    Window)
  else (FLASHW PROMPTWINDOW)
    (NC.PrintMsg NIL T "Can't create global links: graph has no nodes.")]

```

**(NC.BrowserAddLink**

```

[LAMBDA (FromNode ToNode Win Graph GlobalLinkFlg LinkType) ; Edited 28-Sep-88 18:54 by jrc

```

```

;; Like grapher's ADD/AND/DISPLAY/LINK except has different checks and builds a real NC Link.
;; rht 9/20/85: Added GlobalLinkFlg arg to force the link created to be global. Currently, it's global anyway if from node is a sketch card, for
;; example.
;; rht 11/17/85: updated for new card and notefile object format.
;; rht 2/7/86: Now sets and gets browser link labels, etc. via fetch/set fns.
;; rht 1/14/88: Now takes extra LinkType argument and passes to NC.BrowserCreateLink.
;; jrc 8/sep/88: returns link, if one is created.

```

```

(PROG ((MaxDashingStylesNum (LENGTH NC.DashingStyles))
  Link Card LabelPairs LabelPair LabelNum ToNodeID FromNodeID OldDestNode LinkParams ReverseLinkParams
  SavedDeleteLinkFn NumberOfLinks)
(COND
  ([NOT (AND (NC.LinkIconImageObjP (fetch (GRAPHNODE NODELABEL) of FromNode))
    (NC.LinkIconImageObjP (fetch (GRAPHNODE NODELABEL) of ToNode))
    (NC.PrintMsg NIL T "Can't create link from or to a label node. Try 'Add Edge' instead.")
    (FLASHW PROMPTWINDOW)
    (RETURN NIL))
  (SETQ Link (NC.BrowserCreateLink FromNode ToNode Graph Win GlobalLinkFlg LinkType))
    ; We successfully created a link. Now undraw existing link and
    ; redraw with new one added.
    ; First, check whether label for new link is already in graph.
    (SETQ Card (NC.CoerceToCard Win))
    (SETQ LabelPairs (NC.FetchBrowserLinksLegend Card))
    (SETQ LinkType (fetch (Link Label) of Link))
    ;; If link label hasn't appeared in the graph, make a new dashing number for it and update links legend.
    (if (NULL (SETQ LabelPair (FASSOC LinkType LabelPairs)))
      then [SETQ LabelPairs (APPEND LabelPairs (LIST (SETQ LabelPair
        (CONS LinkType (COND
          ((ILESSP (SETQ LabelNum
            (LENGTH LabelPairs)
            ))
            MaxDashingStylesNum)
          (SETQ LabelNum (ADD1
            LabelNum
            )))
          (T LabelNum]
        (NC.SetBrowserLinksLegend Card LabelPairs)
        (NC.MakeLinksLegendMenu Win LabelPairs))
      (SETQ FromNodeID (NC.CoerceToGraphNodeID FromNode))
      (SETQ ToNodeID (NC.CoerceToGraphNodeID ToNode))
      [SETQ NumberOfLinks (PLUS (LENGTH (NC.GraphNodeIDGetProp FromNodeID ToNodeID))
        (LENGTH (NC.GraphNodeIDGetProp ToNodeID FromNodeID))
      (SETQ LinkParams (LINKPARAMETERS FromNode ToNode))
      (SETQ ReverseLinkParams (LINKPARAMETERS ToNode FromNode))
      [COND
        [(OR (ZEROP NumberOfLinks)
          (AND (NULL LinkParams)
            (NULL ReverseLinkParams)))
          (if (OR (FMEMB (fetch (GRAPHNODE NODEID) of FromNode)
            (fetch (GRAPHNODE TONODES) of ToNode))
            (FMEMB (fetch (GRAPHNODE NODEID) of ToNode)
            (fetch (GRAPHNODE TONODES) of FromNode)))
            then ; There are no links, but there is an edge. Delete it and
              ; redisplay.
              (SETQ SavedDeleteLinkFn (fetch (GRAPH GRAPH.DELETELINKFN) of Graph))
              (replace (GRAPH GRAPH.DELETELINKFN) of Graph with NIL)
              (DELETE/AND/DISPLAY/LINK FromNode ToNode Win Graph)
              (NC.PrintMsg NIL T "Replacing existing edge with new link edge.")
              (FLASHW PROMPTWINDOW) ; This is first link between these two nodes so compute dashing
              ; here.
              (replace (GRAPHNODE TONODES) of FromNode with (CONS [LIST LINKPARAMS (fetch (GRAPHNODE NODEID)
                of ToNode)
                'DRAWLINKFN
                (FUNCTION NC.BrowserDrawLinkFn)
                'DASHING
                (CAR (FNTH NC.DashingStyles
                  (CDR LabelPair]
                (fetch (GRAPHNODE TONODES) of FromNode)))
              (replace (GRAPHNODE FROMNODES) of ToNode with (CONS (fetch (GRAPHNODE NODEID) of FromNode)
                (fetch (GRAPHNODE FROMNODES) of ToNode]
                [LinkParams ; There are multiple links. And the link params is on the
                ; FromNode.
                (DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION

```

```

XCOORD _ 0
YCOORD _ 0))
Win Graph 1 LinkParams) ; If there was only one link, then change so multi-link indicator.
(if (EQP NumberOfLinks 1)
  then (RPLACD (CDR LinkParams)
    (LIST 'NODEID FromNodeID 'DESTNODEID ToNodeID]
    ; There are multiple links. The Link params is on the ToNode.
(T (DISPLAYLINK ToNode FromNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Win Graph 1 ReverseLinkParams) ; If there was only one link, then change so multi-link indicator.
(if (EQP NumberOfLinks 1)
  then (RPLACD (CDR ReverseLinkParams)
    (LIST 'NODEID ToNodeID 'DESTNODEID FromNodeID]
(NC.UIDAddProp FromNodeID ToNodeID LabelPair)
[COND
(LinkParams (DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Win Graph 1 LinkParams))
(ReverseLinkParams (DISPLAYLINK ToNode FromNode
(CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Win Graph 1 ReverseLinkParams))
(T (DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Win Graph 1 (LINKPARAMETERS FromNode ToNode])
(NC.PrintMsg NIL T "The " LinkType " link between " (NC.NameOfCardorNodeorImageObj FromNode)
" and "
(NC.NameOfCardorNodeorImageObj ToNode)
" created")
(RETURN NIL])

```

(NC.BrowserAddGlobalLink

```
[LAMBDA (FromNode ToNode Win Graph) (* rht%: "20-Sep-85 14:34")
```

(\* \* This function just allows us to get a GlobalLinkFlg into NC.BrowserAddLink.)

```
(NC.BrowserAddLink FromNode ToNode Win Graph T])
```

(NC.BrowserAddEdgeFn

```
[LAMBDA (Window) ; Edited 15-Dec-88 16:06 by krivacic
```

;;; Called from graph edit menu to create edge.

;;; rht 4/20/87: No longer calls EDITAPPLYTOLINK if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET ((Graph (WINDOWPROP Window 'GRAPH))
AttachW)
  (if (fetch (GRAPH GRAPHNODES) of Graph)
    then (SETQ AttachW (NC.AttachPromptWindow Window))
      (EDITAPPLYTOLINK (FUNCTION NC.BrowserAddEdge)
        'added
        (WINDOWPROP Window 'GRAPH)
        Window AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
      (CLOSEW AttachW)
    else (FLASHW PROMPTWINDOW)
      (NC.PrintMsg NIL T "Can't add edges: graph has no nodes."))

```

(NC.BrowserDeleteCardFn

```
[LAMBDA (Window) ; Edited 15-Dec-88 15:58 by krivacic
```

;;; Called from graph edit menu to delete card and node.

;;; rht 4/20/87: No longer calls NC.BrowserRemoveNode if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
(replace (GRAPH GRAPH.DELETENODEFN) of Graph with NIL)
(if (fetch (GRAPH GRAPHNODES) of Graph)
  then (NC.BrowserRemoveNode Graph Window T)
      (CLOSEW (NC.AttachPromptWindow Window))
  else (FLASHW PROMPTWINDOW)
      (NC.PrintMsg NIL T "Can't delete card/node: graph has no nodes."))

```

(NC.BrowserRemoveNodeFn

[LAMBDA (Window) ; Edited 16-Dec-88 10:01 by krivacic

;; Called from graph edit menu to delete a browser node.

;; rht 4/20/87: No longer calls NC.BrowserRemoveNode if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)]
(replace (GRAPH GRAPH.DELETENODEFN) of Graph with NIL)
(if (fetch (GRAPH GRAPHNODES) of Graph)
then (NC.BrowserRemoveNode Graph Window)
(CLOSEW (NC.AttachPromptWindow Window NIL NIL NIL T))
else (FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "Can't remove node: graph has no nodes."))
```

**(NC.BrowserDeleteLinkFn**

[LAMBDA (Window) ; Edited 16-Dec-88 09:59 by krivacic

;; Called from graph edit menu to delete link and edge.

;; rht 4/20/87: No longer calls EDITMOVENODE if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH))
AttachW]
(if (fetch (GRAPH GRAPHNODES) of Graph)
then (SETQ AttachW (NC.AttachPromptWindow Window))
(EDITAPPLYTOLINK (FUNCTION NC.BrowserDeleteLink)
'deleted
(WINDOWPROP Window 'GRAPH)
Window AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(CLOSEW AttachW)
else (FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "Can't delete link/edge: graph has no nodes."))
```

**(NC.BrowserRemoveEdgeFn**

[LAMBDA (Window) ; Edited 16-Dec-88 10:16 by krivacic

;; Called from graph edit menu to remove browser edge.

;; rht 4/20/87: No longer calls EDITAPPLYTOLINK if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH))
AttachW]
(if (fetch (GRAPH GRAPHNODES) of Graph)
then (SETQ AttachW (NC.AttachPromptWindow Window))
(AND (EDITAPPLYTOLINK (FUNCTION NC.BrowserRemoveEdge)
'removed
(WINDOWPROP Window 'GRAPH)
Window AttachW (FUNCTION NC.NameOfCardorNodeorImageObj))
(NC.MarkCardDirty (NC.CoerceToCard Window)))
(CLOSEW AttachW)
else (FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "Can't remove edges: graph has no nodes."))
```

**(NC.BrowserShrinkLabelFn**

[LAMBDA (Window) ; Edited 15-Dec-88 16:05 by krivacic

;; Shrink a browser label font.

;; rht 4/20/87: No longer calls EDITCHANGEFONT if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)]
(if (fetch (GRAPH GRAPHNODES) of Graph)
then (AND (EDITCHANGEFONT 'SMALLER Window)
(NC.MarkCardDirty (NC.CoerceToCard Window)))
else (FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "Can't shrink label: graph has no nodes."))
```

**(NC.BrowserGrowLabelFn**

[LAMBDA (Window) ; Edited 15-Dec-88 16:05 by krivacic

;; Increase a browser label font.

;;; rht 4/20/87: No longer calls EDITCHANGEFONT if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH]
      (if (fetch (GRAPH GRAPHNODES) of Graph)
          then (AND (EDITCHANGEFONT 'LARGER Window)
                   (NC.MarkCardDirty (NC.CoerceToCard Window)))
          else (FLASHW PROMPTWINDOW)
              (NC.PrintMsg NIL T "Can't expand label: graph has no nodes."))
```

**(NC.BrowserMoveNodeFn**

[LAMBDA (Window) ; Edited 15-Dec-88 16:02 by krivacic

;;; Move a browser node

;;; rht 4/20/87: No longer calls EDITMOVENODE if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH]
      (if (fetch (GRAPH GRAPHNODES) of Graph)
          then (EDITMOVENODE Window)
              (NC.MarkCardDirty (NC.CoerceToCard Window))
          else (FLASHW PROMPTWINDOW)
              (NC.PrintMsg NIL T "Can't move nodes: graph has no nodes."))
```

**(NC.BrowserToggleShadeFn**

```
[LAMBDA (Window)
;; Check if the menu is attached to the Links menu
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH]
      (if (fetch (GRAPH GRAPHNODES) of Graph)
          then (AND (EDITTOGGLELABEL Window)
                   (NC.MarkCardDirty (NC.CoerceToCard Window)))
          else (FLASHW PROMPTWINDOW)
              (NC.PrintMsg NIL T "Can't toggle shade: graph has no nodes."))
```

**(NC.CursorInsideGraphNodeP**

[LAMBDA (GraphNode Window) ; Edited 16-Dec-88 10:09 by krivacic

;; Return non-nil if current cursor position is inside the region of GraphNode.

```
(AND GraphNode (LET ((NodePosition (fetch (GRAPHNODE NODEPOSITION) of GraphNode))
                    (NodeHeight (fetch (GRAPHNODE NODEHEIGHT) of GraphNode))
                    (NodeWidth (fetch (GRAPHNODE NODEWIDTH) of GraphNode)))
                (INSIDEP (CREATEREGION (DIFFERENCE (fetch (POSITION XCOORD) of NodePosition)
                                                  (LRSH NodeWidth 1))
                              (DIFFERENCE (fetch (POSITION YCOORD) of NodePosition)
                                                  (LRSH NodeHeight 1))
                              NodeWidth NodeHeight)
                        (CURSORPOSITION NIL Window]))
```

**(NC.BrowserMoveNodesInRegionFn**

[LAMBDA (Window) ; Edited 15-Dec-88 16:03 by krivacic

;;; Move all the browser nodes in a region.

;;; rht 4/20/87: No longer calls EDITMOVEREGION if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH]
      (if (fetch (GRAPH GRAPHNODES) of Graph)
          then (EDITMOVEREGION Window)
              (NC.MarkCardDirty (NC.CoerceToCard Window))
          else (FLASHW PROMPTWINDOW)
              (NC.PrintMsg NIL T "Can't move nodes: graph has no nodes."))
```

**(NC.BrowserMoveSubtreeFn**

[LAMBDA (Window) ; Edited 15-Dec-88 16:03 by krivacic

;;; Move a browser node and all its descendents.

;;; rht 4/20/87: No longer calls EDITMOVESUBTREE if graph has no nodes.

;; Check if the menu is attached to the Links menu

```
(SETQ Window (OR (WINDOWPROP Window 'MAINWINDOW)
                 Window))
(LET [(Graph (WINDOWPROP Window 'GRAPH)
        (if (fetch (GRAPH GRAPHNODES) of Graph)
            then (EDITMOVESUBTREE Window)
                (NC.MarkCardDirty (NC.CoerceToCard Window))
            else (FLASHW PROMPTWINDOW)
                (NC.PrintMsg NIL T "Can't move nodes: graph has no nodes."))])
```

**(NC.BrowserFixGraphEditMenuFn**

[LAMBDA (Window) ; Edited 23-Jan-89 15:00 by rtk

```
;; Attach the Graph edit menu to the side of the browser window. Put it below the links legend.
;; rht 9/20/85: Now uses ADDMENU offscreen so it doesn't flash in lower left corner before coming up.
;; rht 3/3/86: Now does some 'random' WINDOWPROPS to make reshaping browser not screw up the attached menu.
;; rht 3/7/86: Uses ATTACHMENU to attach the menu.
;; pmi 3/25/87: Added NC.MenuFont to all menus
;; rht 1/14/88: Now copies existing menu hung off window rather than building new one from scratch.
;; bk 1/11/89: Add FIXED-GRAPH-MENU-WINDOW window prop so links menu knows to re-display this menu.
;; bk 1/23/89: Use the LINKSLEGENDWINP prop to fine the links window, set the FIXED-GRAPH-MENU-WINDOW to the window. Close the old
;; fixed menu before fixing a new one. Use the NC.WithTopWindowsUnattached macro to attach the menu.
```

```
(DECLARE (GLOBALVARS NC.MenuFont))
(REMOVEPROMPTWINDOW Window)
(LET* ((ATTACHED-WINDOWS (WINDOWPROP Window 'ATTACHEDWINDOWS))
       [LINKS-MENU-WINDOW (for WINDOW in ATTACHED-WINDOWS thereis (WINDOWPROP WINDOW 'LINKSLEGENDWINP)
                           (OLD-FIXED-MENU (WINDOWPROP WINDOW 'FIXED-GRAPH-MENU-WINDOW))
                           GraphEditMenu GraphEditMenuWin)
       (AND OLD-FIXED-MENU (CLOSEW OLD-FIXED-MENU))
       (REPOSITIONATTACHEDWINDOWS Window)
       (SETQ GraphEditMenu (create MENU using (WINDOWPROP Window 'GRAPHEditUNFIXABLEMENU)
                                           MENUCOLUMN _ 1))
       [if LINKS-MENU-WINDOW
           then (PUTMENUPROP GraphEditMenu 'MAINWINDOW LINKS-MENU-WINDOW)
                (SETQ GraphEditMenuWin (ATTACHMENU GraphEditMenu LINKS-MENU-WINDOW 'BOTTOM 'LEFT))
           else (PUTMENUPROP GraphEditMenu 'MAINWINDOW Window)
                (NC.WithTopWindowsUnattached Window (SETQ GraphEditMenuWin (ATTACHMENU GraphEditMenu Window
                                                                                       'RIGHT
                                                                                       'TOP))
                ; Rig so that close of menu won't close browser.
                (WINDOWDELPROP GraphEditMenuWin 'PASSTOMAINCOMS 'CLOSEW)
                (WINDOWPROP GraphEditMenuWin 'GRAPHEditMENUFLG T)
                (WINDOWPROP Window 'FIXED-GRAPH-MENU-WINDOW GraphEditMenuWin)
                (WINDOWPROP GraphEditMenuWin 'BROWSER-MAINWINDOW Window)
                (WINDOWADDPROP GraphEditMenuWin 'CLOSEFN [FUNCTION (LAMBDA (W)
                                                                    (WINDOWPROP (WINDOWPROP W 'BROWSER-MAINWINDOW)
                                                                    'FIXED-GRAPH-MENU-WINDOW NIL)
                                                                    (DETACHWINDOW W)
                                                                    T))])
```

**(NC.BrowserCreateCard**

[LAMBDA (Graph Window) (\* rht%: "20-Apr-87 15:03")

```
(* Called from grapher ADDNODE fn. Used to create a new card and corresponding browser node.)
(* Rht 11/17/85%: updated for new card and notefile objects.)
(* rht 4/30/86%: Now only computes title if got a link.)
(* rht 4/20/87%: Added ProtectedCardOperation wrapper and call to ALLOW.BUTTON.EVENTS.)
```

```
(LET ((GraphCard (NC.CoerceToCard Window))
      Link GraphNodeID Card Title)
  (NC.ProtectedCardOperation GraphCard "Create card from browser" Window
    (SETQ Link (NC.MakeLink Window NC.BrowserContentsLinkLabel '*New% Card* GraphCard NIL NIL NIL T)
    )
    (if Link
        then [SETQ Title (NC.RetrieveTitle (SETQ Card (fetch (Link DestinationCard) of Link)
                (ALLOW.BUTTON.EVENTS)
                (NC.SetTitle Card (OR (NC.AskUser "Title of new card: " NIL
                                                (if (STREQUAL "Untitled" Title)
                                                    then NIL
                                                    else Title)
                                                T Window)
                "Untitled")) (* Create hash array if haven't already.)
                (NC.GetBrowserHashArray GraphCard Graph)
                (SETQ GraphNodeID (NC.GetBrowserNodeID GraphCard Card))
                (NC.MarkCardDirty GraphCard)
                (NODECREATE GraphNodeID (NC.MakeLinkIcon Link)
                (CURSORPOSITION NIL Window))
                else NIL])
```





```

((NOT LinkAndLabelPair)
(FLASHW PROMPTWINDOW)
(NC.PrintMsg NIL T "No link to delete.")
(RETURN NIL))
((SETQ Link (CAR LinkAndLabelPair))
(NC.ClearMsg Window)
(if (NC.BrowserGetConfirmation (CONCAT " Confirm delete of the " (fetch (Link Label) of Link)
" link between "
(NC.RetrieveTitle SourceCard)
" and "
(NC.RetrieveTitle DestinationCard)
"."
(CHARACTER 13))
NIL Window)
then (NCP.DeleteLinks Link)
(NC.PrintMsg NIL T "The " (fetch (Link Label) of Link)
" link between "
(NC.RetrieveTitle SourceCard)
" and "
(NC.RetrieveTitle DestinationCard)
" deleted")
else (NC.PrintMsg NIL T "Delete Link aborted.")
(RETURN NIL)))
((NOT (NC.YesP (NC.AskUser (CONCAT "Link for that label already deleted." (CHARACTER 13)
"Want to delete its edge? ")
"- -"
'Yes T Window NIL NIL T)))
(RETURN NIL))) ; Undisplay the links between the nodes.
(COND
((SETQ LinkParams (LINKPARAMETERS FromNode ToNode))
(DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Window Graph 1 LinkParams))
((SETQ ReverseLinkParams (LINKPARAMETERS ToNode FromNode))
(DISPLAYLINK ToNode FromNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Window Graph 1 ReverseLinkParams))) ; Remove the first occurrence of the label pair from the proper
; prop list.
[NC.GraphNodeIDPutProp FromNodeID ToNodeID (for RestOfLabelPairs on LabelPairs
bind (LabelPair _ (CADR LinkAndLabelPair))
first (if (EQ LabelPair (CAR LabelPairs))
then (RETURN (CDR LabelPairs)))
do (if (EQ LabelPair (CADR RestOfLabelPairs))
then (RPLACD RestOfLabelPairs (CDDR
RestOfLabelPairs
))
(RETURN LabelPairs)
; If that was the last edge between the two nodes, then remove
; the edge from the graph.
(if (AND (NULL (NC.GraphNodeIDGetProp FromNodeID ToNodeID))
(NULL (NC.GraphNodeIDGetProp ToNodeID FromNodeID)))
then (SETQ FromNodeID (fetch (GRAPHNODE NODEID) of FromNode))
(SETQ ToNodeID (fetch (GRAPHNODE NODEID) of ToNode))
[if LinkParams
then (replace (GRAPHNODE TONODES) of FromNode with (for Node in (fetch (GRAPHNODE TONODES)
of FromNode)
unless (EQ (CADR Node)
ToNodeID)
collect Node))
(replace (GRAPHNODE FROMNODES) of ToNode with (DREMOVE FromNodeID (fetch (GRAPHNODE
FROMNODES)
of ToNode)))
else (replace (GRAPHNODE TONODES) of ToNode with (for Node in (fetch (GRAPHNODE TONODES)
of ToNode)
unless (EQ (CADR Node)
FromNodeID)
collect Node))
(replace (GRAPHNODE FROMNODES) of FromNode with (DREMOVE ToNodeID (fetch (GRAPHNODE
FROMNODES)
of FromNode)
of FromNode]
else ; Redisplay links between the two nodes.
(if LinkParams
then (DISPLAYLINK FromNode ToNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Window Graph 1 LinkParams)
else (DISPLAYLINK ToNode FromNode (CONSTANT (create POSITION
XCOORD _ 0
YCOORD _ 0))
Window Graph 1 ReverseLinkParams])

```

(NC.BrowserAddNode

[LAMBDA (Graph Window)

(\* Randy.Gobbel " 2-Apr-87 15:38")

(\* \* Called by grapher when user creates a new node. Returns new node or nil.)  
 (\* \* rht 11/17/85%: updated to handle new card and notefile formats.)  
 (\* \* pmi 12/5/86%: Modified message to NC.SelectNoteCards to mention SHIFT-selection.  
 Also added GLOBALVARS statement.)  
 (\* \* pmi 12/12/86%: Removed obsolete ReturnLinksFlg argument in call to NC.SelectNoteCards.)  
 (\* \* rg [3/18/87] added NC.ProtectedCardOperation and NC.CardSelectionOperation wrappers)  
 (\* \* rg [4/2/87] changed NC.CardSelectionOperation to NCP.WithLockedCards)

```
(DECLARE (GLOBALVARS NC.BrowserContentsLinkLabel)
(NC.ProtectedCardOperation (NC.CoerceToCard Window)
  "Add Node" Window (NCP.WithLockedCards (PROG ((GraphCard (NC.CoerceToCard Window))
  Link GraphNodeID Card)
    (* Get user to select an existing card.
    Not allowed to create a new one.)
    (SETQ Card (NC.SelectNoteCards
      T
      [FUNCTION (LAMBDA (SelectedCard)
        (COND
          ((NOT (NC.SameCardP
            SelectedCard
            GraphCard))
            T)
          (T (NC.PrintMsg Window T
            "The browser can't
            link to itself."
            (CHARACTER 13)
            "Selection
            ignored."
            (CHARACTER 13))
            NIL]
            NC.SelectingCardMenu GraphCard
            "Shift-select a card or box to include in
            browser."))
    (if (NULL Card)
      then (RETURN NIL))
      (* Make link from browser to new card.)
    (SETQ Link
      (NC.MakeLink Window NC.BrowserContentsLinkLabel Card
      GraphCard NIL NIL NIL T))
      (* Check that it doesn't already exist.
      If not, create a browser node.)
    (COND
      (Link
        (* Create hash array if haven't already.)
        (NC.GetBrowserHashArray GraphCard Graph)
        [SETQ GraphNodeID (NC.GetBrowserNodeID
          GraphCard
          (SETQ Card
            (fetch (Link DestinationCard
              )
              of Link]
            (COND
              ((for Node in (fetch (GRAPH GRAPHNODES)
                of Graph)
                thereis (EQ GraphNodeID (
                  NC.CoerceToGraphNodeID
                  Node)))
                (NC.PrintMsg NIL T "Node for card '"
                  (NC.RetrieveTitle Card)
                  "' already in graph.")
                (FLASHW PROMPTWINDOW)
                (NCP.DeleteLinks Link)
                (RETURN NIL)))
                (NC.MarkCardDirty GraphCard)
                (RETURN (NODECREATE GraphNodeID (
                  NC.MakeLinkIcon
                  Link)
                  (CURSORPOSITION NIL Window])
```

**(NC.BrowserAddLabel**

[LAMBDA (Graph Window)

; Edited 9-Jan-89 16:01 by sye

(\* \* This builds a label node in the default grapher style except that we force non-boxed nodes.)  
 (\* \* pmi 6/23/87%: Added call to NC.MarkCardDirty to make sure substance is marked dirty when nodes are created.)

```
(PROG (NodeLabel NodeName)
  (OR (SETQ NodeLabel (NC.AskUser "Node Label: " NIL NIL T Window T))
    (RETURN))
  (NC.MarkCardDirty (NC.CoerceToCard Window))
  LP (COND
```

```
((FASSOC (SETQ NodeName (PACK* NodeLabel (GENSYM)))
  (fetch (GRAPH GRAPHNODES) of Graph))
 (GO LP)))
(RETURN (NODECREATE NodeName NodeLabel (CURSORPOSITION NIL Window]))
```

**(NC.BrowserAddEdge**

```
[LAMBDA (FromNode ToNode Window Graph) (* rht%: "17-Nov-85 21:45")
```

(\* \* Called by grapher when user adds a new edge. Will only allow edge addition if there are currently no edges between the two nodes.)

```
(LET ((GraphCard (NC.CoerceToCard Window)))
 (COND
  ((EQ FromNode ToNode)
   (NC.PrintMsg NIL T "Can't add edge from a node to itself.")
   (FLASHW PROMPTWINDOW))
  ((OR (MEMBTONODES (fetch (GRAPHNODE NODEID) of ToNode)
    (fetch (GRAPHNODE TONODES) of FromNode))
   (MEMBTONODES (fetch (GRAPHNODE NODEID) of FromNode)
    (fetch (GRAPHNODE FROMNODES) of ToNode)))
   (NC.PrintMsg NIL T "There is already an edge between those two nodes.")
   (FLASHW PROMPTWINDOW))
  (T (NC.MarkCardDirty GraphCard)
   (ADD/AND/DISPLAY/LINK FromNode ToNode Window Graph]))
```

**(NC.BrowserRemoveNode**

```
[LAMBDA (Graph Window DeleteCardFlg NodeToRemove QuietFlg Don'tConfirmFlg)
; Edited 28-Mar-89 12:05 by SYE.ENVOS
```

```
;; Called by grapher when user removes a node.
;; rht 11/17/85: Updated to handle new card and notefile formats.
;; rht 2/7/86: Changed to use NC.SetBrowserRoots instead of NC.PutProp
;; rht 4/30/86: Now takes optional NodeToRemove and QuietFlg argument.
;; rg 3/16/87 NC.DeleteNoteCards -> NC.DeleteNoteCard
;; pmi 8/7/87: Now passes DeleteCardFlg arg to NC.DelBrowserContentsLink. Also turned off confirmation in NC.DeleteNoteCard.
;; rht 1/19/88: Now returns nil if NC.SelectGraphNode returns nil.
;; rht 8/3/88: Ripped out this piece of code that bailed out early in case when NodeToRemove had no edges.
;; pmi & rar 8/26/88: No longer calls FLIPNODE if Window is NIL.
;; kms 3/28/89: Add Don'tConfirmFlg parameter. If the flag is set, no confirmation (for deleting a card) is needed.
;;
```

```
(DECLARE (GLOBALVARS PROMPTWINDOW))
(PROG ((GraphCard (NC.CoerceToCard Window))
  Card CardTitle NodeToRemoveID RootCards NodesToRemove NumVirtuals)
 (OR NodeToRemove [SETQ NodeToRemove (PROG1 (NC.SelectGraphNode Window Graph)
  (if DeleteCardFlg
    then "Choose node of card to delete."
    else "Choose node to remove.")]
  (RETURN NIL))
 (SETQ NodeToRemoveID (NC.CoerceToGraphNodeIDOrLabel NodeToRemove))
 (SETQ Card (NC.CardFromBrowserNodeID NodeToRemoveID))
;; If Don'tConfirmFlg is nil & we're supposed to be deleting a card, then check that node represents a card and that user confirms.
 [OR Don'tConfirmFlg (if DeleteCardFlg
  then (if (NC.ValidCardP Card)
    then (if (NOT (NC.BrowserGetConfirmation (CONCAT " Are you sure you want
  to delete the "
  (SETQ CardTitle
  (NC.RetrieveTitle Card))
  " card? "
  (CHARACTER 13)))
  Graph Window))
    then (AND Window (FLIPNODE NodeToRemove Window))
  (RETURN (NC.PrintMsg NIL T "Card & Node delete
  aborted.)))
  else (NC.PrintMsg NIL T "No card for that node.")
  (FLASHW PROMPTWINDOW)
  (AND Window (FLIPNODE NodeToRemove Window))
  (RETURN NIL))
  else (if (NOT (NC.BrowserGetConfirmation (CONCAT " Confirm Remove of Node "
  (NC.NameOfCardorNodeorImageObj
  NodeToRemove)
  (CHARACTER 13)))
  Graph Window))
    then (NC.PrintMsg NIL T "Node remove aborted.")
  (AND Window (FLIPNODE NodeToRemove Window))
  (RETURN NIL)]
 (AND (NC.ValidCardP GraphCard)
  (NC.MarkCardDirty GraphCard))
```

```
;; We require that all virtual nodes for this node must be removed at once if this one has any link edges to or from it.
```

```
(SETQ NodesToRemove (for Node in (fetch (GRAPH GRAPHNODES) of Graph) when (EQ NodeToRemoveID
(
NC.CoerceToGraphNodeIDOrLabel
Node)))
collect Node)) ; If there are companion virtual nodes, ask for user confirmation.
(if [AND (NOT DeleteCardFlg)
(GREATERP (SETQ NumVirtuals (LENGTH NodesToRemove))
1)
(NOT (PROGN (NC.PrintMsg Window T NumVirtuals " virtual companion nodes will be removed."
(CHARACTER 13))
(NC.YesP (NC.AskUser "Still want to remove? " "--" 'Yes NIL Window NIL NIL T]
then (AND Window (FLIPNODE NodeToRemove Window))
(RETURN NIL)) ; Delete the browsercontents link from the browser.
(AND (NC.ValidCardP Card)
(NC.ValidCardP GraphCard)
(NC.DelBrowserContentsLink GraphCard Card DeleteCardFlg))
; Delete all record of links to and from this node on prop lists.
(NC.SmashGraphNodeIDProps NodeToRemoveID)
(for Node in (fetch (GRAPH GRAPHNODES) of Graph) eachtime (BLOCK) bind UID
when (type? UID (SETQ UID (NC.CoerceToGraphNodeID Node))) do (NC.GraphNodeIDRemProp UID
NodeToRemoveID))
(if (AND (NC.ValidCardP Card)
(NC.ValidCardP GraphCard))
then ; Remove from the root ids list.
(SETQ RootCards (NC.FetchBrowserRoots GraphCard))
(if (for RootCard in RootCards eachtime (BLOCK) thereis (NC.SameCardP Card RootCard))
then (NC.SetBrowserRoots GraphCard (for RootCard in RootCards eachtime (BLOCK)
unless (NC.SameCardP Card RootCard) collect RootCard))
; Remove entry for this node from browser hash array.
)
(NC.RemoveBrowserNodeHashArrayEntry GraphCard Card))
; Get rid of node and its virtual buddies from graph.
(AND Window (FLIPNODE NodeToRemove Window))
(for Node in NodesToRemove do (NC.GraphRemoveNode Node Graph Window T))
; Delete card if we're supposed to.
(if DeleteCardFlg
then (NC.PrintMsg Window T "Deleting the " CardTitle " card " " ... " )
(NC.DeleteNoteCard Card Window T)
(NC.PrintMsg Window NIL "Done." (CHARACTER 13)))
(AND Window (REDISPLAYW Window))
(OR QuietFlg (NC.PrintMsg NIL NIL (CHARACTER 13)
"Node "
(OR CardTitle (NC.NameOfCardorNodeorImageObj NodeToRemove))
(if CardTitle
then " deleted."
else " removed.")))
;; Sye: uncommented this until multiple-node deletion is implemented
;; (OR |QuietFlg| (|NC.PrintMsg| NIL NIL "Nodes: " (|for| |Node| |in| |NodesToRemove| |collect| (PACK* (|NC.NameOfCardorNodeorImageObj|
;; |Node| \))) " removed."))
])
```

(NC.DelBrowserContentsLink

```
[LAMBDA (GraphCard DestCard DeleteCardFlg) (* pmi%: " 7-Aug-87 18:09")
(* * Delete the browsercontents link connecting GraphCard and DestID.)
(* * rht 11/17/85%: updated to handle new card and notefile formats.)
(* * rht 4/30/86%: Now checks to make sure there's a link to delete.)
(* * rht 5/26/87%: Now handles cross-file browsercontents links.)
(* * pmi 8/7/87%: Added DeleteCardFlg arg so we won't create orphans if DestCard is about to be deleted.)
(DECLARE (GLOBALVARS NC.BrowserContentsLinkLabel))
(LET ((Links (NCP.GetLinks GraphCard DestCard NC.BrowserContentsLinkLabel))
(if (CDR Links)
then (NC.ReportError "NC.DelBrowserContentsLink" (CONCAT "Multiple browser contents links between "
(NC.FetchTitle GraphCard)
" and "
(NC.FetchTitle DestCard)))
NIL)
(if Links
then (* Don't create an orphan if this card is about to be deleted.)
(NC.DeleteLink (CAR Links)
DeleteCardFlg T)
else (* Could be that it's a cross file browsercontents link.)
(for Link in (NCP.GetLinks GraphCard NIL NC.BrowserContentsLinkLabel)
bind CrossFileLinkCard (DestCardUID _ (fetch (Card UID) of DestCard))
when [AND (NC.CrossFileLinkCardP (SETQ CrossFileLinkCard (fetch (Link DestinationCard)
of Link)))
(NC.SameUIDP DestCardUID (fetch (CrossFileLinkSubstance CrossFileLinkDestCardUID)
of (NCP.CardSubstance CrossFileLinkCard]
do (RETURN (NC.DeleteLink Link T T]))
```



```

                                unless (EQ (CADR Node)
                                           ToNodeID)
                                collect Node)
      (replace (GRAPHNODE FROMNODES) of ToNode with (DFIRSTREMOVE FromNodeID
                                                    (fetch (GRAPHNODE FROMNODES)
                                                          of ToNode)))
    else (replace (GRAPHNODE TONODES) of ToNode with (for Node in (fetch (GRAPHNODE TONODES)
                                                                           of ToNode)
                                                         unless (EQ (CADR Node)
                                                             FromNodeID)
                                                         collect Node)
              (replace (GRAPHNODE FROMNODES) of FromNode with (DFIRSTREMOVE ToNodeID
                                                                    (fetch (GRAPHNODE FROMNODES)
                                                                          of FromNode]
                    ; Redisplay links between the two nodes.
    else
      (NC.DisplayGraphLinksBetween FromNode ToNode Window Graph])

```

**(NC.BrowserChangeLabel**

```

[LAMBDA (GRAPH W NODE)
  (if (NC.LinkIconImageObjP (fetch (GRAPHNODE NODELABEL) of NODE))
      then (NC.PrintMsg NIL T "Cannot change label to a card node" (CHARACTER 13))
          (FLASHW PROMPTWINDOW)
      else (NC.AskUser "Node Label: " NIL NIL T W T])

```

**(NC.BrowserGetConfirmation**

```

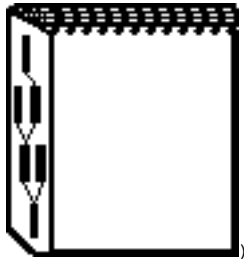
[LAMBDA (PROMPT GRAPH WINDOW)
  (MOUSECONFIRM PROMPT NIL (GETPROMPTWINDOW WINDOW])

```

; Edited 13-Dec-88 16:23 by krivacic

;;; Stuff for shrinking graphs and browsers.

(RPAQQ NC.GraphIconBM



(RPAQQ NC.GraphIconMaskBM



(RPAQQ NC.GraphCardIcon



(DEFINEQ

**(NC.GraphCardShrinkFn**

```

[LAMBDA (W)

```

; Edited 2-Dec-88 14:56 by krivacic

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

;; rg 11/19/86 Added GLOBALVARS declaration

;; rht 1/21/88: Now checks windowprop to see whether default shrunken icon template should be overridden.

```

(DECLARE (GLOBALVARS TEDIT.ICON.FONT NC.ShrunkenGraphIconTemplate NC.IconProps))
(* LET ((OldIconTitle (WINDOWPROP W
  (QUOTE GRAPH.ICON.TITLE)))
  (Icon (WINDOWPROP W (QUOTE ICON)))
  (Card (NC.CoerceToCard W) IconTitle BM IconTemplate
  BMHeight BMHalfHeight TitleRegion)
  (SETQ IconTitle (CONCAT "NC:"
  (NC.RetrieveTitle Card))) (COND
  ((NOT Icon) (WINDOWPROP W (QUOTE ICON)
  (SETQ Icon (TILEDICONW (OR
  (WINDOWPROP W (QUOTE ShrunkenIconTemplate))
  (AND (WINDOWPROP W (QUOTE NoteCardObject))
  (LISTGET NC.IconProps (fetch (CardObject Type) of

```

```
(WINDOWPROP W (QUOTE NoteCardObject))))))
NC.ShrunkenGraphIconTemplate IconTitle TEDIT.ICON.FONT
(POSITIONP (WINDOWPROP W
(QUOTE SHRUNKENWINPOS))) T))))))
(COND ((NOT (EQUAL OldIconTitle IconTitle))
(WINDOWPROP W (QUOTE GRAPH.ICON.TITLE) IconTitle)
(ICONTITLE IconTitle TitleRegion NIL Icon))))
```

```
:: Generic Shrink Window
(NC.ShrinkIcon W)
```

)

;;; Stuff for making copies of graphs for making imageobj's.

(DEFINEQ

**(NC.MakeExternalGraphCopy**

[LAMBDA (Graph)

(\* rht%: "24-Mar-86 15:50")

(\* Make a copy of Graph in which each link icon nodelabel has been replaced by a copy of itself and MoveNodeFn, etc. are replaced by NIL.)

(\* rht 3/24/86%: Now creates new graphnodes via "using" instead of "copying" to preserve EQness of UIDs.)

```
(create GRAPH
DIRECTEDFLG _ (fetch (GRAPH DIRECTEDFLG) of Graph)
SIDESFLG _ (fetch (GRAPH SIDESFLG) of Graph)
GRAPHNODES _ (for GraphNode in (fetch (GRAPH GRAPHNODES) of Graph) bind Box NewNode Label
collect (LET (NewLinkIcon)
(PROG1 [SETQ NewNode (create GRAPHNODE
using GraphNode NODELABEL _
(if (NC.LinkIconImageObjP
(SETQ Label (fetch (GRAPHNODE NODELABEL)
of GraphNode)))
then (SETQ NewLinkIcon (NC.LinkIconCopyFn
Label))
else (COPY Label]
(if NewLinkIcon
then (SETQ Box (NC.ExternalPutLinkIconImageBoxFn NewLinkIcon NIL
NIL NIL))
(replace (GRAPHNODE NODEHEIGHT) of NewNode
with (fetch (IMAGEBOX YSIZE) of Box))
(replace (GRAPHNODE NODEWIDTH) of NewNode
with (fetch (IMAGEBOX XSIZE) of Box))))))
```

)

;;; Add the Graph card to the cardtypes list.

(DEFINEQ

**(NC.AddGraphCard**

[LAMBDA NIL

; Edited 25-Feb-88 11:37 by Randy.Gobbel

;;; fgh 11/14/85: Updated to conform to merge of substance and card types.

;;; fgh 2/17/86 Added attached bitmap field.

;;; rht 4/7/86: Added middle button menu items.

;;; pmi 5/26/87: Changed default card size as requested by Cathy Marshall.

;;; dwm 1/6/88: Added ReadOnly/ReadWrite FUNCTIONS

```
(NC.AddCardType 'Graph 'NoteCard `[(MakeFn , (FUNCTION NC.MakeGraphCard))
(EditFn , (FUNCTION NC.BringUpGraphCard))
(QuitFn , (FUNCTION NC.GraphCardCloseFn))
(MakeReadOnlyFn , (FUNCTION NC.MakeGraphCardReadOnly))
(MakeReadWriteFn , (FUNCTION NC.MakeGraphCardReadWrite))
(GetFn , (FUNCTION NC.GetGraphSubstance))
(PutFn , (FUNCTION NC.PutGraphSubstance))
(CopyFn , (FUNCTION NC.GraphCopySubstance))
(MarkDirtyFn , (FUNCTION NC.MarkGraphDirty))
(DirtyPFn , (FUNCTION NC.GraphDirtyP))
(CollectLinksFn , (FUNCTION NC.CollectReferencesInGraph))
(DeleteLinksFn , (FUNCTION NC.DelReferencesToCardFromGraph))
(UpdateLinkIconsFn , (FUNCTION NC.UpdateLinkImagesInGraph))
(InsertLinkFn , (FUNCTION NC.InsertLinkInGraph))
(TranslateWindowPositionFn , (FUNCTION
NC.TranslateWindowPositionToGraphPosition]
`((LinkDisplayMode Title)
(DisplayedInMenuFlg ,T)
(DefaultWidth 500)
(DefaultHeight 330)
```



```
(LinkAnchorModesSupported T)
(LinkIconAttachedBitMap ,NC.GraphCardIcon)
(MiddleButtonMenuItems ,'((Relayout% Graph (FUNCTION NC.RelayoutGraphCard)
                          "Re-layout the graph, but keep same nodes.")]
)

(DECLARE%: DONTEVAL@LOAD

(NC.AddGraphCard)
)

(PUTPROPS NCGRAPHCARD FILETYPE :FAKE-COMPILE-FILE)

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

(PUTPROPS NCGRAPHCARD COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1988 1989 1990 1993 1994 2020))
```

FUNCTION INDEX

NC.AddGraphCard	32	NC.GetGraphEditMenu	18
NC.BringUpGraphCard	4	NC.GetGraphSubstance	7
NC.BrowserAddEdge	28	NC.GraphAddLabel	12
NC.BrowserAddEdgeFn	21	NC.GraphAddLinkFn	11
NC.BrowserAddGlobalLink	21	NC.GraphAddNode	11
NC.BrowserAddLabel	27	NC.GraphAddNodeFn	10
NC.BrowserAddLabelFn	19	NC.GraphCardCloseFn	6
NC.BrowserAddLink	20	NC.GraphCardEditCommandFn	15
NC.BrowserAddNode	26	NC.GraphCardEditEventFn	15
NC.BrowserAddNodeFn	19	NC.GraphCardLeftButtonFn	6
NC.BrowserChangeLabel	31	NC.GraphCardMiddleButtonFn	6
NC.BrowserChangeLabelFn	19	NC.GraphCardShrinkFn	31
NC.BrowserCreateCard	24	NC.GraphChangeLabelFn	9
NC.BrowserCreateCardFn	18	NC.GraphCopySubstance	6
NC.BrowserCreateGlobalLinkFn	19	NC.GraphDeleteLinkFn	11
NC.BrowserCreateLink	25	NC.GraphDeleteNodeFn	11
NC.BrowserCreateLinkFn	19	NC.GraphDirtyP	7
NC.BrowserDeleteCardFn	21	NC.GraphEditMenuWhenSelectedFn	15
NC.BrowserDeleteLink	25	NC.GRAPHERCOPYBUTTONEVENTFN	9
NC.BrowserDeleteLinkFn	22	NC.GraphFontChangeFn	11
NC.BrowserFixGraphEditMenuFn	24	NC.GraphMoveNodeFn	11
NC.BrowserGetConfirmation	31	NC.GraphNodeIDGetProp	14
NC.BrowserGrowLabelFn	22	NC.GraphNodeIDPutProp	14
NC.BrowserMoveNodeFn	23	NC.GraphNodeIDRemProp	14
NC.BrowserMoveNodesInRegionFn	23	NC.GraphNodeLinkIconUpdate	7
NC.BrowserMoveSubtreeFn	23	NC.GraphRemoveNode	12
NC.BrowserRemoveEdge	30	NC.InsertEdgeInGraph	13
NC.BrowserRemoveEdgeFn	22	NC.InsertLabelInGraph	13
NC.BrowserRemoveNode	28	NC.InsertLinkInGraph	14
NC.BrowserRemoveNodeFn	21	NC.InsideGrapherNodeP	10
NC.BrowserRightButtonFn	18	NC.MakeExternalGraphCopy	32
NC.BrowserShrinkLabelFn	22	NC.MakeGraphCard	4
NC.BrowserToggleShadeFn	23	NC.MakeGraphCardReadOnly	9
NC.CardFromBrowserNodeID	17	NC.MakeGraphCardReadWrite	9
NC.CoerceToGraphNodeID	16	NC.MarkGraphDirty	7
NC.CoerceToGraphNodeIDOrLabel	17	NC.PutGraphSubstance	7
NC.CollectReferencesInGraph	5	NC.RelayoutGraphCard	5
NC.CursorInsideGraphNodeP	23	NC.SaveOrRevertGraphCard	9
NC.DelBrowserContentsLink	29	NC.SelectGraphNode	17
NC.DelReferencesToCardFromGraph	8	NC.SetupGraphEditMenus	18
NC.DisplayGraphLinksBetween	17	NC.SmashGraphNodeIDProps	17
NC.GCDirected/UndirectedFn	16	NC.TrackCursorInGrapherWindow	13
NC.GCFixGraphEditMenuFn	16	NC.TranslateWindowPositionToGraphPosition	14
NC.GCSetupGraphEditMenus	15	NC.UpdateLinkImagesInGraph	8
NC.GCSides/CentersFn	16	NCAddStub.GraphCard	3
NC.GCToggleBoxFn	16		

VARIABLE INDEX

NC.GCGraphEditMenuItems	3	NC.GraphCardIcon	31	NC.GraphIconMaskBM	31
NC.GCGraphEditUnfixedMenuItems	3	NC.GraphIconBM	31		

PROPERTY INDEX

NCGRAPHCARD	33
-------------	----