File created: 3-Nov-2020 16:13:45 **{DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCTYPESMECH.;5**

changes to: (FNS NC.AddCardType)

previous date: 9-Jan-94 19:34:28 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCTYPESMECH.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ NCTYPESMECHCOMS
       (

;;; Internal variables

        (GLOBALVARS NC.CardTypes NC.TypesLock)
        (INITVARS (NC.CardTypes NIL)
               (NC.TypesLock (CREATE.MONITORLOCK "NoteCards Type Mechanism")))


;;; Note Cards Type Mechanism

        (FNS NC.IsSubTypeOfP NC.CardTypeRecord NC.MakeCardTypesList NC.AddCardType NC.AddCardTypeStub
             NC.ModifyCardType NC.RecomputeCardType NC.ListOfCardTypes NC.SubTypesOfCardType NC.DeleteCardType
             NC.InheritFieldsFromSuperType NC.PropagateCardTypeFieldsDownward NC.AutoLoadCardType
             NC.CardTypeLoader NC.CardTypeStubLoader NC.GetCardTypeFieldOfSuper)
        (BITMAPS NC.TypelessIcon)
        (DECLARE%: DONTEVAL@LOAD (P (NC.MakeCardTypesList)))
        (PROP (FILETYPE MAKEFILE-ENVIRONMENT)
              NCTYPESMECH)))


;;; Internal variables

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.CardTypes NC.TypesLock)
)

(RPAQ? NC.CardTypes NIL)

(RPAQ? NC.TypesLock (CREATE.MONITORLOCK "NoteCards Type Mechanism"))


;;; Note Cards Type Mechanism

(DEFINEQ

(NC.IsSubTypeOfP
  [LAMBDA (SubTypeName SupposedSuperTypeName)                   (* rht%: "15-Jul-86 18:06")

             (* * Is SubTypeName the name of a Card type that is a sub type of carrd type SupposedSuperTypeName.)

    (LET (SuperType)
         (OR (EQ SubTypeName SupposedSuperTypeName)
             (AND (SETQ SuperType (NC.GetCardTypeField SuperType SubTypeName))
                  (NC.IsSubTypeOfP SuperType SupposedSuperTypeName])

(NC.CardTypeRecord
  [LAMBDA (TypeName NewTypeRecord)                              (* kirk%: "12-Feb-86 16:15")

             (* * Retrieve the card type record for TypeName)

             (* * fgh |1/31/86| Updated to handle hash table for card types. Also added code to allow modification of type record.
             Does not do any consstency checking.)

    (PROG1 (GETHASH TypeName NC.CardTypes)
        (if NewTypeRecord
            then (PUTHASH (fetch (NoteCardType TypeName) of NewTypeRecord)
                        NewTypeRecord NC.CardTypes)))])

(NC.MakeCardTypesList
  [LAMBDA NIL                                                   (* Randy.Gobbel "24-Nov-87 15:32")

             (* * Make initial set of card types.)

             (* * fgh |1/31/86| Updated to handle hash table for card types.)
```

```
                (* * fgh |2/17/86| Added typless attached bit map to NoteCards card type.)

                (* * rht 4/7/86%: Now sets up default left and middle button menu items.)

                (* * rg |3/27/87| added WITH.MONITOR)

                (* * rg |11/24/87| fn fields now initialized to NILL (instead of FILLME))

    (DECLARE (GLOBALVARS NC.TypelessIcon NC.CardTypes NC.DefaultLeftButtonMenuItems
                    NC.DefaultMiddleButtonMenuItems NC.TypesLock NC.NoteCardTypeMenu))
    (WITH.MONITOR NC.TypesLock
        (SETQ NC.CardTypes (OR (HARRAYP NC.CardTypes)
                               (HASHARRAY 50)))
        (PUTHASH 'NoteCard
               (create NoteCardType
                      TypeName _ 'NoteCard
                      SuperType _ NIL
                      StubFlg _ NIL
                      MakeFn _ (FUNCTION NILL)
                      EditFn _ (FUNCTION NILL)
                      QuitFn _ (FUNCTION NILL)
                      MakeReadOnlyFn _ (FUNCTION NILL)
                      MakeReadWriteFn _ (FUNCTION NILL)
                      GetFn _ (FUNCTION NILL)
                      PutFn _ (FUNCTION NILL)
                      CopyFn _ [FUNCTION (LAMBDA (Card ToStream FromStream Length)
                                                                    (* Copy a card's substance using copybytes.)
                                           (LET* ((FromStartPtr (GETFILEPTR FromStream))
                                                  (FromEndPtr (PLUS Length FromStartPtr)))
                                                 (COPYBYTES FromStream ToStream FromStartPtr FromEndPtr)
                                                 T]
                      MarkDirtyFn _ (FUNCTION NILL)
                      DirtyPFn _ (FUNCTION NILL)
                      CollectLinksFn _ (FUNCTION NILL)
                      DeleteLinksFn _ (FUNCTION NILL)
                      UpdateLinkIconsFn _ (FUNCTION NILL)
                      InsertLinkFn _ (FUNCTION NILL)
                      TranslateWindowPositionFn _ (FUNCTION NILL)
                      LinkDisplayMode _ (create LINKDISPLAYMODE
                                               ATTACHBITMAPFLG _ T)
                      DefaultWidth _ 100
                      DefaultHeight _ 100
                      LinkAnchorModesSupported _ NIL
                      DisplayedInMenuFlg _ NIL
                      LinkIconAttachedBitMap _ NC.TypelessIcon
                      LeftButtonMenuItems _ NC.DefaultLeftButtonMenuItems
                      MiddleButtonMenuItems _ NC.DefaultMiddleButtonMenuItems)
               NC.CardTypes)
        (SETQ NC.NoteCardTypeMenu))])

(NC.AddCardType
  [LAMBDA (TypeName SuperType FnsAssocList VarsAssocList FullDefinitionFile ListOfFILLMEFields StubFlg
                 SystemGeneratedCard)                                    ; Edited 12-Jan-89 11:25 by rtk

;;; Create a new note card type and link it into the card type heirarchy.

;;; fgh 1/31/86 Updated to handle card type hash table.

;;; fgh 2/3/86 Updated to do inheritance at definition time rather than at access time.

;;; rht 7/15/86: Innards replaced by call to NC.ModifyCardType.

;;; rht 11/7/86: Now accepts ListOfFILLMEFields arg.  Any such fields get the \\FILLME// atom as value.  No longer forces load of super.  Takes three
;;; extra optional args StubFlg, FullDefinitionFile, and ListOfFILLMEFields.

;;; RG 3/27/87 added WITH.MONITOR

;;; BK 1/11/89: Add SystemGeneratedCard parameter for the CardTypes menu.

    (if SystemGeneratedCard
        then (PUTPROP TypeName 'SYSTEM-GENERATED T))
    (WITH.MONITOR NC.TypesLock
        (LET [NewType (MainFieldNames (CONSTANT (for FieldName in (RECORDFIELDNAMES 'NoteCardType)
                                                   when (NEQ 'InheritedFlg (SUBATOM FieldName -12 -1))
                                                   collect FieldName]
             (COND
                ((NULL TypeName)
                 (NC.ReportError "NC.AddCardType" "Illegal type name: NIL"))
                ((NULL SuperType)
                 (NC.ReportError "NC.AddCardType" "Illegal super type name: NIL")))

;;; Create new NoteCardType

             (SETQ NewType (create NoteCardType
                                  TypeName _ TypeName
                                  SuperType _ SuperType
                                  StubFlg _ StubFlg
                                  FullDefinitionFile _ FullDefinitionFile))

;;; If it's a stub, then change the default values for fields from \\FILLME// to \\EMPTY// except for those in ListOfFILLMEFields.

             [if StubFlg
                 then (for FieldName in MainFieldNames unless (FMEMB FieldName ListOfFILLMEFields)
                         when (EQ (RECORDACCESS `(NoteCardType ,FieldName)
                                         NewType NIL 'fetch)
                                   '\\FILLME//)
                         do (RECORDACCESS `(NoteCardType ,FieldName)
                                   NewType NIL 'replace '\\EMPTY//]

;;; Compute fields, inheriting from above and propagating downward.

             (NC.ModifyCardType NewType FnsAssocList VarsAssocList)

;;; Stash the new card type record, reset the menu and then return.

             (PUTHASH TypeName NewType NC.CardTypes)
             (SETQ NC.NoteCardTypeMenu)
             NewType))])
```

```
(NC.AddCardTypeStub
  [LAMBDA (TypeName SuperType FullDefinitionFileName FnsAssocList VarsAssocList ListOfFILLMEFields)
                                                                 (* rht%: " 7-Nov-86 14:55")

          (* * Create a card type stub to hold as a place holder until the actual card type is autoloaded.)

          (* * If FullDefinitionFileName is not supplied, one is constructed from the type name.)

          (* * kirk 18Jun86 Changed to use FullDefinitionFileName and deal with NIL or UnLoaded SuperType.
          Now gets called recursively in that case. Hence, the above warning.)

          (* * fgh |8/26/86| Revamped. Now attempts to load supertype stub or whole definition if not already loaded.
          Must be called with valid SuperType arg.)

          (* * rht 11/7/86%: Now just calls NC.AddCardType with StubFlg = T.)

    (NC.AddCardType TypeName SuperType FnsAssocList VarsAssocList FullDefinitionFileName ListOfFILLMEFields T])
```

```
(NC.ModifyCardType
  [LAMBDA (TypeRecord FnsAssocList VarsAssocList)                ; Edited 15-Dec-88 10:59 by krivacic

          (* * Changes the fields of the given type, inheriting other fields from the super and propagating changes downward.
          This used to be the innards of NC.AddCardType.)

    (if (type? NoteCardType TypeRecord)
        then [LET [(MainFieldNames (CONSTANT (for FieldName in (RECORDFIELDNAMES 'NoteCardType)
                                                when (NEQ 'InheritedFlg (SUBATOM FieldName -12 -1)) collect
                                                                                                        FieldName
                                                 ]
                  (for AssocPair in (APPEND FnsAssocList VarsAssocList) when (FMEMB (CAR AssocPair)
                                                                                   MainFieldNames)
                    do

          (* * Set the proper field to the given value)

                        (RECORDACCESS `(NoteCardType ,(CAR AssocPair))
                               TypeRecord NIL 'replace (OR (AND (FMEMB (CAR AssocPair)
                                                                       '(LeftButtonMenuItems
                                                                               MiddleButtonMenuItems))
                                                                (NC.ExpandBars (CADR AssocPair)))
                                                           (CADR AssocPair)))

          (* * Set the corresponding InheritedFlg to indicate that this field is not inherited.)

                        (RECORDACCESS `[NoteCardType ,(PACK* (CAR AssocPair)
                                                             'InheritedFlg]
                               TypeRecord NIL 'replace NIL)
                        (AND (EQ (CAR AssocPair)
                                 'LinkIconAttachedBitMap)
                             (NC.AddCardIconToIconProps (fetch (NoteCardType TypeName) of TypeRecord)
                                    (CADR AssocPair]

          (* * Inherit any NIL fields from super type record.)

             (AND (fetch (NoteCardType SuperType) of TypeRecord)
                  (NC.InheritFieldsFromSuperType TypeRecord))

          (* * Propagate any changes downward to the subtypes of this type.)

             (NC.PropagateCardTypeFieldsDownward TypeRecord])
```

```
(NC.RecomputeCardType
  [LAMBDA (TypeName FnsAssocList VarsAssocList)                    (* Randy.Gobbel "27-Mar-87 17:05")

             (* * Recomputes an existing card type, changing those fields present on FnsAssocList and VarsAssocList.)

             (* * rg |3/27/87| added WITH.MONITOR)

    (WITH.MONITOR NC.TypesLock
        (LET ((TypeRecord (NC.CardTypeRecord TypeName))
               SuperType)
              [COND
                 ((NULL TypeRecord)
                  (NC.ReportError "NC.RecomputeCardType" (CONCAT "Unknown type: " TypeName)))
                 ([AND (NEQ TypeName 'NoteCard)
                       (NULL (NC.CardTypeRecord (SETQ SuperType (fetch (NoteCardType SuperType) of TypeRecord]
                  (NC.ReportError "NC.RecomputeCardType" (CONCAT TypeName " has unknown super type:  " SuperType]

             (* * Compute fields, inheriting from above and propagating downward.)

              (NC.ModifyCardType TypeRecord FnsAssocList VarsAssocList)

             (* * reset the menu)

              (SETQ NC.NoteCardTypeMenu)
              TypeName))])

(NC.ListOfCardTypes
  [LAMBDA (ForMenuFlg)                                              (* fgh%: " 2-Sep-86 00:30")

             (* * Return a list of the note card types)

             (* * Special processing for older types to make them not all caps.)

             (* * fgh |1/31/86| Updated to handle card type Hash array.)

             (* * fgh |9/1/86| Updated to handle \\EMPTY// slots in CardType defns.)

    (SORT (LET (CollectLst)
                [MAPHASH NC.CardTypes (FUNCTION (LAMBDA (TypeRecord TypeName)
                                                  (if (OR (NULL ForMenuFlg)
                                                          (EQ (fetch (NoteCardType DisplayedInMenuFlg) of
                                                                                                     TypeRecord
                                                                    )
                                                              T))
                                                      then (push CollectLst (fetch (NoteCardType TypeName)
                                                                                 of TypeRecord]
                CollectLst])

(NC.SubTypesOfCardType
  [LAMBDA (TypeName)                                                (* fgh%: "31-Jan-86 21:49")

             (* * Returns a list of the types names of all the sub-types of TypeName)

             (* * fgh |1/31/86| First created.)

    (LET (CollectLst)
         [MAPHASH NC.CardTypes (FUNCTION (LAMBDA (TypeRecord Key)
                                           (if (EQ TypeName (fetch (NoteCardType SuperType) of TypeRecord))
                                               then (push CollectLst (fetch (NoteCardType TypeName) of TypeRecord]
         CollectLst])

(NC.DeleteCardType
  [LAMBDA (TypeName DeleteSubTypesFlg)                              (* pmi%: "10-Dec-87 11:23")

             (* * Deletes a card type. If DeleteSubTypesFlg is non-NIL recursively deletes all sub-types.
             If DeleteSubTypesFlg is NIL, then attempting to delete a type with sub-types is an error.)

             (* * fgh |1/31/86| First created.)

             (* * rg |3/27/87| added WITH.MONITOR)

             (* * dsj 9/27/87%: now clears NC.NoteCardTypeMenu so that the type no longer appears on the menu of card types.)

             (* * pmi 12/10/87%: Added dsj's change; see above comment.)

    (DECLARE (GLOBALVARS NC.NoteCardTypeMenu))
    (WITH.MONITOR NC.TypesLock
        (LET ((SubTypes (NC.SubTypesOfCardType TypeName)))
              [if SubTypes
                  then (if DeleteSubTypesFlg
                           then (for SubType in SubTypes do (NC.DeleteCardType SubType DeleteSubTypesFlg))
                         else (NC.ReportError NIL (CONCAT "Cannot delete a card type with existing sub-types."
                                                           (CHARACTER 13)
```

```
                                                     "This type has sub-types: " SubTypes]
          (PUTHASH TypeName NIL NC.CardTypes)
          (SETQ NC.NoteCardTypeMenu)))])

(NC.InheritFieldsFromSuperType
  [LAMBDA (TypeNameOrRecord SuperTypeRecord)                          ; Edited  4-Dec-87 12:44 by rht:

         (* * Inherit fields from super card type record if the supertype is not a stub.)

         (* * fgh |2/3/86| First created.)

         (* * fgh |8/26/86| Now uses SuperTypeRecord consistently if it is present.)

         (* * rht 11/7/86%: Changed procedure for deciding when to inherit a field from upstairs.)

    (LET [(TypeRecord (if (type? NoteCardType TypeNameOrRecord)
                          then TypeNameOrRecord
                        else (NC.CardTypeRecord TypeNameOrRecord)))
          (InheritedFlgNames (CONSTANT (for FieldName in (RECORDFIELDNAMES 'NoteCardType)
                                          when (EQ 'InheritedFlg (SUBATOM FieldName -12 -1)) collect FieldName]
         (if [OR SuperTypeRecord (SETQ SuperTypeRecord (NC.CardTypeRecord (fetch (NoteCardType SuperType)
                                                                             of TypeRecord]
             then (for FlgName in InheritedFlgNames
                     do (LET ((FieldName (SUBATOM FlgName 1 -13))
                              FieldValue SuperFieldValue)
                             (SETQ FieldValue (RECORDACCESS '(NoteCardType ,FieldName)
                                                     TypeRecord NIL 'fetch))
                             (SETQ SuperFieldValue (RECORDACCESS '(NoteCardType ,FieldName)
                                                          SuperTypeRecord NIL 'fetch))

         (* * Only inherit from the super field if it's got a real value, and if either the subfield has value \\FILLME// or the InheritedFlg
         field says to inherit. Don't inherit if the subfield is \\EMPTY//, cause that indicates a field for a stub that shouldn't inherit.)

                             (if [AND [NOT (FMEMB SuperFieldValue '(\\FILLME// \\EMPTY//]
                                      (NOT (EQ FieldValue '\\EMPTY//))
                                      (OR (EQ FieldValue '\\FILLME//)
                                          (RECORDACCESS '(NoteCardType ,FlgName)
                                                 TypeRecord NIL 'fetch]
                                 then (RECORDACCESS '(NoteCardType ,FieldName)
                                             TypeRecord NIL 'replace SuperFieldValue])

(NC.PropagateCardTypeFieldsDownward
  [LAMBDA (TypeNameOrRecord)                                          (* rht%: " 7-Nov-86 14:54")

         (* * For all subtypes of TypeNameOrRecord, inherit fields from TypeNameOrRecord.)

         (* * fgh |2/3/86| First created.)

         (* * rht 11/7/86%: Fixed minor typo.)

    (for SubType in (NC.SubTypesOfCardType (if (type? NoteCardType TypeNameOrRecord)
                                                then (fetch (NoteCardType TypeName) of TypeNameOrRecord)
                                              else TypeNameOrRecord))
       do (NC.InheritFieldsFromSuperType SubType (if (type? NoteCardType TypeNameOrRecord)
                                                      then TypeNameOrRecord
                                                    else NIL))
          (NC.PropagateCardTypeFieldsDownward SubType])

(NC.AutoLoadCardType
  [LAMBDA (TypeName FieldName FileSuggestion)                         ; Edited  4-Dec-87 12:44 by rht:

         (* * AutoLoad card type TypeName using FileSuggestion as the first place to look.
         Otherwise look in the directories specified by NoteCardsDirectories.
         Works by starting at the give Type and moving up the inheritance heirarchy loading all types that are currently defined only
         by stubs until it reaches a card type for which FieldName is defined.
         This works because full inheritance is done whereever possible during type definition in NC.AddCardType.)

         (* * fgh |2/3/86| First created.)

         (* * fgh |7/16/86| Added ALLOW.BUTTON.EVENTS call)

         (* * kirk |8/22/86| Added check for NIL card type)

    (ALLOW.BUTTON.EVENTS)
    (LET ((CardTypeRecord (NC.CardTypeRecord TypeName))
          (OriginalTypeName TypeName)
          FieldValue)

         (* * If there's no stub for this card type, load the type definition file.)

         (if (NULL TypeName)
             then (NC.ReportError NIL "NIL card type"))
         [if (NULL CardTypeRecord)
             then (NC.CardTypeLoader TypeName FileSuggestion)
          (* * Still no definition then we're sunk.)

                  (if (NULL (SETQ CardTypeRecord (NC.CardTypeRecord TypeName)))
                      then (NC.ReportError NIL (CONCAT "Cannot find definition of card type: " TypeName]

          (* * We have at least a stub for this card type. Move up the inheritance hierarchy loading the real files instead of the stubs
          until we have a value of the field we are looking for.)

         (until (EQ TypeName 'NoteCard) do

          (* * If this is a stub, load the real thing. Break if it can't be found.)

                                                  (if (fetch (NoteCardType StubFlg) of CardTypeRecord)
                                                      then (OR (NC.CardTypeLoader TypeName FileSuggestion)
                                                               (NC.ReportError NIL (CONCAT "Cannot find full definition
                                                                                      of card type: " TypeName)))
                                                           (SETQ CardTypeRecord (NC.CardTypeRecord TypeName)))

          (* * If the field was defined in the real thing, then return it. Else keep moving up load ing all the stubs.)

                                                  (if (SETQ FieldValue (RECORDACCESS `(NoteCardType ,FieldName)
                                                                              CardTypeRecord NIL 'fetch))
                                                      then (RETURN FieldValue)
                                                    else (SETQ TypeName (fetch (NoteCardType SuperType) of CardTypeRecord))
                                                         (SETQ CardTypeRecord (NC.CardTypeRecord TypeName)))
            finally (NC.ReportError NIL (CONCAT "Cannot find definition of card type: " OriginalTypeName "   Field
                                                       name needed: " FieldName])

(NC.CardTypeLoader
  [LAMBDA (TypeName FileSuggestion QuietFlg)                           ; Edited 20-Jan-88 11:33 by Randy.Gobbel

;;; Load card type TypeName using FileSuggestion as the first place to look.  Otherwise look in the directories specified by NoteCardsDirectories.

;;; fgh 2/3/86 First written.

;;; kirk 1Jul86 Added new process.

;;; kirk 15Jul86 Removed new process and added FGH's RESETLST to close stream.

;;; fgh 7/16/86 Added PRINTOUT to inform user before find file begins.

;;; kirk 8/7/86 Added QuietFlg.  Will not stop LOAD messages though.

;;; kirk 19/8/86 Added set QuietFlg T in OPENWP check

;;; fgh 8/26/86 Changed LOAD?  to a LOAD.  If we are at this point there must bve some reason we need to reload the file even if its alreadt been
;;; loaded.

;;; rht&rg&pmi 10/21/86: Changed LOAD to FILESLOAD.

;;; rht 4/17/87: Now smashes PAGEFULLFN of tty window so that it'll scroll automatically.

;;; rg 1/20/88 rewritten to call NC.LoadFileFromDirectories, eliminating lots of identical code

    (NC.LoadFileFromDirectories FileSuggestion NIL QuietFlg TypeName])

(NC.CardTypeStubLoader
  [LAMBDA (TypeName FileSuggestion)                                    ; Edited 21-Jan-88 13:59 by Randy.Gobbel

;;; Load card type TypeName using FileSuggestion as the first place to look.  Otherwise look in the directories specified by NoteCardsDirectories.

;;; fgh 2/3/86 First written.

;;; kirk 6/20/86 Added NCP.Add...CardStub fn load and call

;;; kirk 9/9/86 Changed the name of the FN to search for to NCAddStub...Card

;;; kirk 9/21/86 added RESETLST

;;; rg 10/21/86: Changed to use simpler scheme for finding files

;;; rht&rg 11/7/86: Now passes (QUOTE VARS) arg to LOADFNS so that interesting vars like link icon bitmap will be loaded.  There ought to be a better
;;; way.

;;; rg 5/18/87 checks, via disgusting kludge, to see if file can be opened random access, if not, loads whole file

;;; rht 6/12/87: Changed call to OPENSTREAM to open for INPUT rather than BOTH.

;;; rg 6/18/87 now checks for files on NS servers by looking for a colon in the file name

;;; rg 6/19/87 check for AddTypeFn already defined before attempting to load it

;;; rg&pmi 9/8/87: fixes stub loading for NS servers.
;;; rg 1/21/88 removed special case for NS servers, call to NC.FixFileDates

    (LET* ((FileName (NC.FindFile FileSuggestion TypeName T))
           (oldFILEDATES (GETPROP FileName 'FILEDATES))
           AddTypeFn)
          (if FileName
              then (RESETLST
                       [RESETSAVE NIL `(COND
                                          (oldFILEDATES (PUTPROP ,(LIST 'QUOTE FileName)
                                                               'FILEDATES oldFILEDATES))
                                          (T (REMPROP ,(LIST 'QUOTE FileName)
                                                     'FILEDATES]

;;; Load and eval the NC.Add...CardStub fn

                       (SETQ AddTypeFn (PACK* 'NCAddStub. TypeName 'Card))
                       (OR (FNTYP AddTypeFn)
                           (LOADFNS AddTypeFn FileName NIL 'VARS)))
                   (if (NOT (FNTYP AddTypeFn))
                       then (NC.ReportError "NC.CardTypeStubLoader" (CONCAT "NoteCards: Can't find card type stub
                                                                        init procedure: " AddTypeFn " in "
                                                                        FileName ".")))

;;; this should call NC.AddCardTypeStub

                  (APPLY AddTypeFn])

(NC.GetCardTypeFieldOfSuper
  [LAMBDA (CardTypeName FieldName)                                ; Edited  4-Dec-87 12:44 by rht:

          (* * Get the specifdied field from the super of CardTypeName)

          (* * fgh |8/26/86| First created as companion for MACRO NC.GetCardTypeField.)

    (EVAL `(NC.GetCardTypeField ,FieldName (fetch (NoteCardType SuperType) of (NC.CardTypeRecord CardTypeName])

)

(RPAQQ NC.TypelessIcon ▯)

(DECLARE%: DONTEVAL@LOAD

(NC.MakeCardTypesList)
)

(PUTPROPS NCTYPESMECH FILETYPE :FAKE-COMPILE-FILE)

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

(PUTPROPS NCTYPESMECH COPYRIGHT ("Venue & Xerox Corporation" 1984 1985 1986 1987 1988 1989 1990 1993 1994
                                         2020))
```

{MEDLEY}<notecards>system>NCTYPESMECH.;1   28-Jun-2024 18:35:17

-- Listed on 30-Jun-2024 13:57:11 --

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX