

File created: 5-Nov-2020 19:33:38 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCHASHCARD.;5

changes to: (FNS NCAddStub.HashCard NC.AddHashCard NC.HashCardMakeFn NC.HashCardEditFn NC.HashCardGetFn  
NC.HashCardPutFn NC.HashCardCopyFn NC.AddRegistryCard NC.RegistryCardMakeFn  
NC.RegistryCardGetFn NC.RegistryCardPutFn NC.RegisterCardByName NC.LookupCardByName  
NC.UnregisterName)

previous date: 9-Jan-94 19:47:17 {DSK}<users>arunwelch>skydrive>documents>unix>lisp>lde>notecards>system>NCHASH  
CARD.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

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

(RPAQQ NCHASHCARDOMS  
(

::: Stuff for the hash card type. This includes the basic general hash card and a specialization called Registry card type.

```
(FNS NCAddStub.HashCard)
(GLOBALVARS NC.DefaultHashCardHashArraySize)
(INITVARS (NC.DefaultHashCardHashArraySize 100))
(FNS NC.AddHashCard NC.HashCardMakeFn NC.HashCardEditFn NC.HashCardGetFn NC.HashCardPutFn
NC.HashCardCopyFn)
(DECLARE%: DONTEVAL@LOAD (P (NC.AddHashCard)))
```

::: This is a specialization of Hash card called Registry card. It maps atoms to UIDs and has tailored Get and Put fns.

```
(GLOBALVARS NC.RegistrySubstanceEndMarker NC.DefaultRegistryCardHashArraySize)
(INITVARS (NC.DefaultRegistryCardHashArraySize 100)
(NC.RegistrySubstanceEndMarker '###EndRegistrySubstance##))
(FNS NC.AddRegistryCard NC.RegistryCardMakeFn NC.RegistryCardGetFn NC.RegistryCardPutFn)
(FNS NC.RegisterCardByName NC.LookupCardByName NC.UnregisterName)
(DECLARE%: DONTEVAL@LOAD (P (NC.AddRegistryCard)))
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
NCHASHCARD))
```

::: Stuff for the hash card type. This includes the basic general hash card and a specialization called Registry card type.

(DEFINEQ

**(NCAddStub.HashCard**

[LAMBDA NIL

(\* kirk%: "19-Jun-86 20:52")

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

(NC.AddCardTypeStub 'Hash 'NoteCard 'NCHASHCARD])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.DefaultHashCardHashArraySize)

)

(RPAQ? NC.DefaultHashCardHashArraySize 100)

(DEFINEQ

**(NC.AddHashCard**

[LAMBDA NIL

; Edited 3-Dec-87 19:01 by rht:

(\* \* rht 7/14/86%: No longer has a QuitFn.)

```
(NC.AddCardType 'Hash 'NoteCard `[(MakeFn ,(FUNCTION NC.HashCardMakeFn))
(EditFn ,(FUNCTION NC.HashCardEditFn))
(GetFn ,(FUNCTION NC.HashCardGetFn))
(PutFn ,(FUNCTION NC.HashCardPutFn))
(CopyFn ,(FUNCTION NC.HashCardCopyFn))
(MarkDirtyFn ,(FUNCTION NIL))
(DirtyPFn ,(FUNCTION NIL))
(QuitFn ,(FUNCTION NIL))
'((LinkAnchorModesSupported NIL])
```

**(NC.HashCardMakeFn**

[LAMBDA (Card Title NoDisplayFlg HASHARRAYArgs)

(\* rht%: "26-Feb-86 10:21")

(\* \* Make a hash substance. HASHARRAYArgs should be a list of arguments to the HASHARRAY call. HASHARRAYArgs should be a list the first element of which is a positive integer. Otherwise use default.)

```
(if [OR (NOT (LISTP HASHARRAYArgs))
      (NOT (AND (FIXP (CAR HASHARRAYArgs))
                (GREATERP (CAR HASHARRAYArgs)
                           0]))
      then (SETQ HASHARRAYArgs (LIST NC.DefaultHashCardHashArraySize)))
(NC.SetSubstance Card (APPLY (FUNCTION HASHARRAY)
                             HASHARRAYArgs))
Card])
```

**(NC.HashCardEditFn**

```
[LAMBDA NIL
(NC.ReportError NIL "Cannot edit a hash substance"')] (* rht%: "26-Feb-86 10:22")
```

**(NC.HashCardGetFn**

```
[LAMBDA (Card Length Stream) (* kirk%: "16-Sep-86 17:34")
(* * Get the hash substance from the disk)
(* * |9/16/86| changed READ to HREAD)
(HREAD Stream)]
```

**(NC.HashCardPutFn**

```
[LAMBDA (Card Stream) (* kirk%: "16-Sep-86 17:33")
(* * Puts hash substance to notefile.)
(* * kirk |9/16/84| changed PRINT to HPRINT)
(HPRINT (NC.FetchSubstance Card)
Stream)
0])
```

**(NC.HashCardCopyFn**

```
[LAMBDA (Card FromStream ToStream Length) (* rht%: "26-Feb-86 10:24")
(* * Copy a hash substance from FromStream to ToStream.)
(LET* ((FromStartPtr (GETFILEPTR FromStream))
       (FromEndPtr (PLUS Length FromStartPtr)))
(COPYBYTES FromStream ToStream FromStartPtr FromEndPtr))
T])
```

)

(DECLARE%: DONTEVAL@LOAD

**(NC.AddHashCard)**

)

;;; This is a specialization of Hash card called Registry card. It maps atoms to UIDs and has tailored Get and Put fns.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.RegistrySubstanceEndMarker NC.DefaultRegistryCardHashArraySize)

)

(RPAQ? NC.DefaultRegistryCardHashArraySize 100)

(RPAQ? NC.RegistrySubstanceEndMarker '###EndRegistrySubstance##)

(DEFINEQ

**(NC.AddRegistryCard**

```
[LAMBDA NIL
(NC.AddCardType 'Registry 'Hash `[(MakeFn ,(FUNCTION NC.RegistryCardMakeFn))
                                       (GetFn ,(FUNCTION NC.RegistryCardGetFn))
                                       (PutFn ,(FUNCTION NC.RegistryCardPutFn))
                                       '((LinkAnchorModesSupported NIL))]
; Edited 3-Dec-87 19:01 by rht:
```

**(NC.RegistryCardMakeFn**

```
[LAMBDA (Card Title NoDisplayFlg) (* rht%: "26-Feb-86 11:09")
(* * Make a Registry substance.)
(NC.ApplySupersFn MakeFn Card Title NoDisplayFlg (LIST NC.DefaultRegistryCardHashArraySize])
```

**(NC.RegistryCardGetFn**

[LAMBDA (Card Length Stream) (\* rht%: " 1-Nov-86 16:06")

(\* \* Get the Registry substance from the disk)

(\* \* rht 11/1/86%: Now uses our readtable when reading.)

```
(DECLARE (GLOBALVARS NC.OrigReadTable NC.RegistrySubstanceEndMarker NC.DefaultRegistryCardHashArraySize))
(LET [(HashArray (HASHARRAY NC.DefaultRegistryCardHashArraySize))
      (EndLoc (PLUS Length (GETFILEPTR Stream)
                      EndLoc))
      (for bind Key eachtime (BLOCK) while (LESSP (GETFILEPTR Stream)
                                                  EndLoc))
      until (EQ (SETQ Key (READ Stream NC.OrigReadTable))
              NC.RegistrySubstanceEndMarker)
      do
        (BIN Stream)
        (PUTHASH Key (NC.ReadUID Stream)
                  HashArray))
      HashArray])
```

(\* Skip CR)

**(NC.RegistryCardPutFn**

[LAMBDA (Card Stream) (\* rht%: " 1-Nov-86 16:08")

(\* \* Puts Registry substance to notefile. Writes down atomic key followed by UID for each hash table pair. Writes down special marker at the end.)

(\* \* rht 11/1/86%: Now uses our readtable when printing.)

```
(DECLARE (GLOBALVARS NC.OrigReadTable NC.RegistrySubstanceEndMarker))
[MAPHASH (NC.FetchSubstance Card)
  (FUNCTION (LAMBDA (Value Item)
            (PRINT Item Stream)
            (NC.WriteUID Stream Value))
  (PRINT NC.RegistrySubstanceEndMarker Stream NC.OrigReadTable)
  0])
```

)

(DEFINEQ

**(NC.RegisterCardByName**

[LAMBDA (RegistryCard Name Card) (\* rht%: "26-Feb-86 15:38")

(\* \* Stuff the item/val pair Name/Card into RegistryCard's hash array. Note that RegistryCard should be active when this is called.)

```
(PUTHASH (MKATOM Name)
  (fetch (Card UID) of Card)
  (NC.FetchSubstance RegistryCard))
(NC.MarkCardDirty RegistryCard])
```

**(NC.LookupCardByName**

[LAMBDA (RegistryCard Name) (\* fgh%: " 2-May-86 22:25")

(\* \* Look up in RegistryCard's hash array the card hash'ed by key Name. Note that RegistryCard must be active when this is called.)

(\* \* fgh |5/2/86| Now handles case where no match is found in registry table!)

```
(LET [(UIDFound (GETHASH (MKATOM Name)
                        (NC.FetchSubstance RegistryCard))
      (if UIDFound
          then (NC.CardFromUID UIDFound (fetch (Card NoteFile) of RegistryCard))
```

**(NC.UnregisterName**

[LAMBDA (RegistryCard Name) (\* rht%: "26-Feb-86 15:39")

(\* \* Remove any entry for Name from RegistryCard's hash array. Note that RegistryCard should be active when this is called.)

```
(PUTHASH (MKATOM Name)
  NIL
  (NC.FetchSubstance RegistryCard))
(NC.MarkCardDirty RegistryCard])
```

)

(DECLARE%: DONTEVAL@LOAD

**(NC.AddRegistryCard)**

)

(PUTPROPS NCHASHCARD FILETYPE :FAKE-COMPILE-FILE)

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

Page 4

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

(PUTPROPS **NHASHCARD COPYRIGHT** ("Venue & Xerox Corporation" 1986 1987 1988 1989 1990 1993 1994 2020))

---

**FUNCTION INDEX**

NC.AddHashCard .....	1	NC.HashCardGetFn .....	2	NC.RegisterCardByName ...	3	NC.UnregisterName .....	3
NC.AddRegistryCard .....	2	NC.HashCardMakeFn .....	1	NC.RegistryCardGetFn ....	3	NCAddStub.HashCard .....	1
NC.HashCardCopyFn .....	2	NC.HashCardPutFn .....	2	NC.RegistryCardMakeFn ...	2		
NC.HashCardEditFn .....	2	NC.LookupCardByName .....	3	NC.RegistryCardPutFn ....	3		

---

**VARIABLE INDEX**

NC.DefaultHashCardHashArraySize .....	1	NC.RegistrySubstanceEndMarker .....	2
NC.DefaultRegistryCardHashArraySize .....	2		

---

**PROPERTY INDEX**

NCHASHCARD ....., 3, 4

---