

File created: 26-May-86 18:04:19 {QV}<NOTECARDS>1.3K>LIBRARY>NCSTAT.;8

changes to: (FNS NCStat.InfoAboutFileBoxHierarchy NCStat.MarkDepthAndTerminals NCStat.SizeOfFileBoxHierarchy
NCStat.DepthOfFileBoxHierarchy NCStat.#FileBoxesInHierarchy NCStat.#LinksFromCard
NCStat.#CardsInNoteFile)
(VARS NCSTATCOMS)

previous date: 22-May-86 17:55:47 {QV}<NOTECARDS>1.3K>LIBRARY>NCSTAT.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(* * Copyright (c) 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ NCSTATCOMS ((FNS NCStat.SizeOfFileBoxHierarchy NCStat.#LinksFromCard NCStat.InfoAboutFileBoxHierarchy
NCStat.MarkDepthAndTerminals NCStat.#CardsInNoteFile)))

(DEFINEQ

(NCStat.SizeOfFileBoxHierarchy

[LAMBDA (RootBox)

(* fgh: "22-May-86 20:09")

(* * How many file boxes and cards in the file box hierarchy that starts at RootCard.
Include both SubBox and Filedcard links to help very early NFs e.g., Nato-missiles, that had a problem distinguishing these
links.)

(* * fgh 5/22/86 First created.)

(if (NCP.ValidCard RootBox)
then (LET ((Closure (NCP.ComputeTransitiveClosure RootBox (QUOTE (SubBox FiledCard))
999999))
(LIST (for Card in Closure count (EQ (QUOTE FileBox)
(NCP.CardType Card))
(for Card in Closure count (NEQ (QUOTE FileBox)
(NCP.CardType Card))

(NCStat.#LinksFromCard

[LAMBDA (Card Don'tIncludeGlobalFlg)

(* fgh: "22-May-86 17:56")

(* * Returns the number of ToLinks emanating from Card. If Don'tIncludeGlobalFlg is non-NIL, eliminates Global ToLinks

(* * fgh 5/22/86 First created.)

(LET (Links)
(if (NULL Card)
then (SETQ Card (NCP.WhichCard)))
[if (NCP.ValidCard Card)
then (SETQ Links (NCP.GetLinks Card))
(if Don'tIncludeGlobalFlg
then (SETQ Links (for Link in Links when (NEQ (QUOTE GLOBAL)
(CAADR (NCP.LinkDesc Link)))
collect Links]
(LENGTH Links])

(NCStat.InfoAboutFileBoxHierarchy

[LAMBDA (RootCard)

(* fgh: "26-May-86 17:01")

(* * Compute some interesting info about a filebox hierarchy)

(* * fgh 5/22/86 First created.)

(LET
(Marker (GENSYM))
(TerminalMarker (GENSYM))
(if (NCP.ValidCard RootCard)
then (NCStat.MarkDepthAndTerminals RootCard 0 Marker TerminalMarker)
(PROG1 (LIST (QUOTE #FileBoxes)
(LET ((Count 0))
[NCP.MapCards (fetch (Card NoteFile) of RootCard)
(FUNCTION (LAMBDA (Card)
(if (AND (EQ (NCP.CardType Card)
(QUOTE FileBox))
(NC.FetchUserDataProp Card Marker))
then (SETQ Count (ADD1 Count))
Count)
(QUOTE #Cards)
(LET ((Count 0))
[NCP.MapCards (fetch (Card NoteFile) of RootCard)
(FUNCTION (LAMBDA (Card)
(if (AND (NEQ (NCP.CardType Card)
(QUOTE FileBox))
(NC.FetchUserDataProp Card Marker))

```

                                then (SETQ Count (ADD1 Count]
Count)
(QUOTE MaxDepth)
(LET ((MaxDepth 0))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (LET (Depth)
        (if [AND (EQ (NCP.CardType Card)
                     (QUOTE FileBox))
              (SETQ Depth (CAR (NC.FetchUserDataProp Card
                               Marker]
                            then (SETQ MaxDepth (MAX MaxDepth Depth]
MaxDepth)
(QUOTE MinDepth)
(LET ((MinDepth 9999999))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (LET (Depth)
        (if (AND (EQ (NCP.CardType Card)
                     (QUOTE FileBox))
              (SETQ Depth (CAR (NC.FetchUserDataProp Card
                               Marker)))
          (NC.FetchUserDataProp Card TerminalMarker))
        then (SETQ MinDepth (MIN MinDepth Depth]
MinDepth)
(QUOTE AvgDepth)
(LET ((SumDepth 0)
      (Count 0))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (LET (Depth)
        (if (AND (SETQ Depth (CAR (NC.FetchUserDataProp Card
                               Marker)))
              (NC.FetchUserDataProp Card TerminalMarker))
          then (SETQ SumDepth (PLUS SumDepth Depth))
              (SETQ Count (ADD1 Count]
(FQUOTIENT SumDepth Count))
(QUOTE AvgBoxesPerBox)
(LET ((Sum 0)
      (Count 0))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (if (EQ (NCP.CardType Card)
              (QUOTE FileBox))
        then (LET (#SubBoxes)
              (if (SETQ #SubBoxes (CADDR (NC.FetchUserDataProp
                                         Card Marker)))
                then (SETQ Sum (PLUS Sum #SubBoxes))
                    (SETQ Count (ADD1 Count]
(FQUOTIENT Sum Count))
(QUOTE MaxCardsPerBox)
(LET ((Max 0))
  [NCP.MapCards
    (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (if (EQ (NCP.CardType Card)
              (QUOTE FileBox))
        then (LET (#Cards)
              (if (SETQ #Cards (CADDR (NC.FetchUserDataProp Card
                               Marker)))
                then (SETQ Max (MAX Max #Cards))
                    (if (ZEROP #Cards)
                      then (PRINT (NCP.CardTitle Card]
Max)
(QUOTE MinCardsPerBox)
(LET ((Min 999999))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (if (EQ (NCP.CardType Card)
              (QUOTE FileBox))
        then (LET (#Cards)
              (if (SETQ #Cards (CADDR (NC.FetchUserDataProp
                                         Card Marker)))
                then (SETQ Min (MIN Min #Cards]
Min)
(QUOTE MinCardsPerBoxForBoxesWithGT1Card)
(LET ((Min 999999))
  [NCP.MapCards (fetch (Card NoteFile) of RootCard)
    (FUNCTION (LAMBDA (Card)
      (if (EQ (NCP.CardType Card)
              (QUOTE FileBox))
        then (LET (#Cards)
              (if (AND (SETQ #Cards (CADDR (
                                         NC.FetchUserDataProp
                                         Card Marker)))
                    (NOT (ZEROP #Cards)))
                then (SETQ Min (MIN Min #Cards]

```

```

    Min)
  (QUOTE AvgCardsPerBox)
  (LET ((Sum 0)
        (Count 0))
    [NCP.MapCards (fetch (Card NoteFile) of RootCard)
      (FUNCTION (LAMBDA (Card)
        (if (EQ (NCP.CardType Card)
              (QUOTE FileBox))
            then (LET (#Cards)
                  (if (SETQ #Cards (CADDR (NC.FetchUserDataProp
                                           Card Marker)))
                    then (SETQ Sum (PLUS Sum #Cards))
                        (SETQ Count (ADD1 Count))

                    (FQUOTIENT Sum Count))
              (QUOTE AvgBoxesPerCard)
            (LET ((Sum 0)
                  (Count 0))
              [NCP.MapCards (fetch (Card NoteFile) of RootCard)
                (FUNCTION (LAMBDA (Card)
                  (if (NEQ (NCP.CardType Card)
                          (QUOTE FileBox))
                    then (LET (#Cards)
                          (if (SETQ #Cards (CADR (NC.FetchUserDataProp
                                                    Card Marker)))
                            then (SETQ Sum (PLUS Sum #Cards))
                                (SETQ Count (ADD1 Count))

                            (FQUOTIENT Sum Count))
                      (QUOTE #CardsFiledInMoreThan1Box)
                    (LET ((Count 0))
                      [NCP.MapCards (fetch (Card NoteFile) of RootCard)
                        (FUNCTION (LAMBDA (Card)
                          (if (NEQ (NCP.CardType Card)
                                  (QUOTE FileBox))
                            then (if (AND (CADR (NC.FetchUserDataProp Card Marker))
                                          (GREATERP (CADR (NC.FetchUserDataProp Card
                                                            Marker))
                                                    1))
                                then (SETQ Count (ADD1 Count))

                                Count)
                          (QUOTE #BoxesFiledInMoreThan1Box)
                        (LET ((Count 0))
                          [NCP.MapCards (fetch (Card NoteFile) of RootCard)
                            (FUNCTION (LAMBDA (Card)
                              (if (EQ (NCP.CardType Card)
                                      (QUOTE FileBox))
                                then (if (AND (CADR (NC.FetchUserDataProp Card Marker))
                                              (GREATERP (CADR (NC.FetchUserDataProp Card
                                                                Marker))
                                                        1))
                                  then (SETQ Count (ADD1 Count))

                                  Count))
                            (NCP.MapCards (fetch (Card NoteFile) of RootCard)
                              (FUNCTION (LAMBDA (Card)
                                (NC.SetUserDataProp Card Marker NIL)
                                (NC.SetUserDataProp Card TerminalMarker NIL))]))])

```

(NCStat.MarkDepthAndTerminals

```

[LAMBDA (RootCard Depth Marker TerminalMarker) (* fgh: "22-May-86 21:30")
```

(* Mark RootCard for its depth in hierarchy, then recurse on the children of root.)

(* fgh 5/22/86 First created.)

```

(LET (Children)
  [NC.SetUserDataProp RootCard Marker (LIST Depth (LENGTH (NCP.CardParents RootCard))
      (if (EQ (NCP.CardType RootCard)
              (QUOTE FileBox))
        then (for Card in (NCP.FileBoxChildren RootCard)
              count (EQ (NCP.CardType Card)
                        (QUOTE FileBox))
              (if (EQ (NCP.CardType RootCard)
                      (QUOTE FileBox))
                then (for Card in (NCP.FileBoxChildren RootCard)
                      count (NEQ (NCP.CardType Card)
                                (QUOTE FileBox))
                (if [SETQ Children (NCP.GetChildren RootCard (QUOTE (SubBox FiledCard))
                    then (for Card in Children do (NCStat.MarkDepthAndTerminals Card (ADD1 Depth)
                                                Marker TerminalMarker))
                (if [AND (EQ (NCP.CardType RootCard)
                          (QUOTE FileBox))
                    (for Card in Children never (EQ (NCP.CardType Card)
                                                    (QUOTE FileBox))
                    then (SETQ XXC (ADD1 XXC))
                        (NC.SetUserDataProp RootCard TerminalMarker T])

```

(NCStat.#CardsInNoteFile

[LAMBDA (NoteFile TypeList)

(* fgh: "22-May-86 17:53")

(* * Returns the number of cards of one of the types in TypeList in NoteFile.)

(* * fgh 5/22/86 First created.)

```
(if TypeList
  then (SETQ TypeList (MKLIST TypeList)))
(if (type? NoteFile NoteFile)
  then (LET ((Count 0)
             [NCP.MapCards NoteFile (FUNCTION (LAMBDA (Card)
                                               (if (AND (NCP.ValidCard Card)
                                                    (OR (NULL TypeList)
                                                        (FMEMB (NCP.CardType Card)
                                                                TypeList))))
                                               then (SETQ Count (ADD1 Count]
                                               Count])
```

)

(PUTPROPS **NCSTAT COPYRIGHT** ("Xerox Corporation" 1986))

FUNCTION INDEX

NCStat.#CardsInNoteFile4 NCStat.InfoAboutFileBoxHierarchy .1 NCStat.SizeOfFileBoxHierarchy1
NCStat.#LinksFromCard1 NCStat.MarkDepthAndTerminals3

VARIABLE INDEX

NCSTATCOMS1
