

File created: 13-Dec-87 22:41:04 {EG:PARC:XEROX}<RAO>LISP>USERS>NCDRAFTCARD.;5

changes to: (FNS ncdraft-draft-pre-expand-fn)

previous date: 28-Nov-87 17:54:39 {EG:PARC:XEROX}<RAO>LISP>USERS>NCDRAFTCARD.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1987 by Xerox Corporation. All rights reserved.

(RPAQQ NCDRAFTCARDCOMS
(

::: NCDRAFTCARD - an extensible document generator.

; Document Style Manipulation

```
(RECORDS ncdraft-style)
(FNS ncdraft-props-to-style ncdraft-style-to-props)
[DECLARE%: DONTEVAL@LOAD (VARS (NC.CRString (CONCAT (CHARACTER 13)))
                                (NC.DocTitleParaLeading 20))
  (INITVARS (*ncdraft-user-props*))
  (VARS (*ncdraft-default-props* ' (BuildBackLinks NONE CopyEmbeddedLinks NONE NumberEmbeddedLinks
                                   NONE ExpandEmbeddedLinks NONE LocateEmbeddedLinks NONE
                                   TitleEmbeddedLinks NONE SectionStart (1)
                                   SectionCRs 2 SectionReferencePrefix "Section "
                                   SectionBoldP T InsureSectionCRs 2 TitleCRs 2
                                   InsureTitleCRs 2 BibRefEmbeddedLinks NONE
                                   ProcessSubstanceP T))
    (*ncdraft-default-style*))
  (P (SETQ *ncdraft-default-style* (ncdraft-props-to-style *ncdraft-user-props* (
                                                                    ncdraft-props-to-style
                                                                    *ncdraft-default-props*
                                                                    (create
                                                                    ncdraft-style
                                                                    ]
```

:: Document Generation by Traversal

```
(RECORDS ncdraft-state)
(FNS ncdraft-do-make-draft ncdraft-traverse-and-map-cards ncdraft-traverse-and-map-exportable-card
    ncdraft-traverse-and-map-text-card)
(FNS ncdraft-edit-draft-style ncdraft-edit-draft-style-with-inspector ncdraft-recompute-draft)
```

:: Draft Card Specific Methods

```
(FNS ncdraft-draft-pre-card-fn ncdraft-draft-process-segment-fn ncdraft-draft-pre-expand-fn
    ncdraft-draft-post-expand-fn ncdraft-process-exportable-fn)
(FNS ncdraft-section-down-level ncdraft-section-up-level ncdraft-section-to-string)
```

:: Latex Interface Functions.

```
(VARIABLES **NCDRAFT-LATEX-P** **NCDRAFT-LATEX-CITE-P**)
(FUNCTIONS LATEX-SECTION LATEX-SECTION-PTR LATEX-CITE CONVERT-BIB-STRING)
```

:: Tedit Interface Functions. First set used. Second set not used.

```
(GLOBALVARS NC.DocBackPtrLinkLabel NC.SubBoxLinkLabel NC.FiledCardLinkLabel PSA.Database
    NC.SelectingCardMenu NC.CRString NC.DocTitleParaLeading)
(VARIABLES **NCDRAFT-COLLECT-BIB-P**)
(INITVARS (*nc-draft-bib-refs*))
(FNS ncdraft-append-string-to-stream ncdraft-insure-crs)
(FNS ncdraft-fetch-to-links-in-order ncdraft-add-cr-if-needed ncdraft-change-para-leading)
(FNS PUT-BIB-REFS)
```

:: User Interface to Document Style Editing

```
[VARS (NC.DocumentStyleEditSpec ' ((ExpandLinks ExpandEmbeddedLinks ALL NONE Select)
                                   (BackToCards BuildBackLinks ALL NONE SelCard)
                                   (CopyLinks CopyEmbeddedLinks ALL NONE Select)
                                   (TitleLinks TitleEmbeddedLinks ALL NONE Select)
                                   (SectionLinks NumberEmbeddedLinks ALL NONE Select)
                                   (ToSectionsLinks LocateEmbeddedLinks ALL NONE Select)
                                   (ToBibLinks BibRefEmbeddedLinks ALL NONE Select])
  (FNS ncdraft-do-edit-draft-style ncdraft-fetch-draft-style-field ncdraft-select-draft-style-field
    ncdraft-ask-card-type)
```

:: Register Card Type

```
(FILES NCTEXTCARD)
(FNS ncdraft-make-draft ncdraft-get-draft ncdraft-put-draft)
(FNS NC.AddDraftCard)
(P (NC.AddDraftCard))
(FNS NCAddStub.DraftCard)
```

::

```
(DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT NCDRAFTCARD)
```

(PROP FILETYPE NCDRAFTCARD)))))

;;; NCDRAFTCARD - an extensible document generator.

;; Document Style Manipulation

(DECLARE%: EVAL@COMPILE

(DATATYPE ncdraft-style (BuildBackLinks CopyEmbeddedLinks TitleEmbeddedLinks NumberEmbeddedLinks
ExpandEmbeddedLinks LocateEmbeddedLinks SectionStart SectionCRs
SectionReferencePrefix SectionBoldP TitleBoldP InsureSectionCRs TitleCRs
InsureTitleCRs BibRefEmbeddedLinks ProcessSubstanceP TabSectionP))
)

(/DECLAREDATATYPE 'ncdraft-style
'(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---

' 34)

(DEFINEQ

(ncdraft-props-to-style

[LAMBDA (props draft-style) (* Rao "22-Mar-87 14:06")
(for field on props by (CDDR field) do (RECORDACCESS (LIST 'ncdraft-style (CAR field))
draft-style NIL 'REPLACE (CADR field)))
draft-style])

(ncdraft-style-to-props

[LAMBDA (style) (* Rao "22-Mar-87 14:08")
(for field in (RECORDFIELDNAMES 'ncdraft-style) join (LIST field (RECORDACCESS (LIST 'ncdraft-style field)
style))
)

(DECLARE%: DONTEVAL@LOAD

(RPAQ NC.CRString (CONCAT (CHARACTER 13)))

(RPAQQ NC.DocTitleParaLeading 20)

(RPAQ? *ncdraft-user-props*)

(RPAQQ *ncdraft-default-props*

(BuildBackLinks NONE CopyEmbeddedLinks NONE NumberEmbeddedLinks NONE ExpandEmbeddedLinks NONE
LocateEmbeddedLinks NONE TitleEmbeddedLinks NONE SectionStart (1)
SectionCRs 2 SectionReferencePrefix "Section " SectionBoldP T InsureSectionCRs 2 TitleCRs 2
InsureTitleCRs 2 BibRefEmbeddedLinks NONE ProcessSubstanceP T))

(RPAQQ *ncdraft-default-style* NIL)

[SETQ *ncdraft-default-style* (ncdraft-props-to-style *ncdraft-user-props* (ncdraft-props-to-style
ncdraft-default-props
(create ncdraft-style]
)

;; Document Generation by Traversal

(DECLARE%: EVAL@COMPILE

(DATATYPE ncdraft-state (DocCard DocStream DocObj Section PreCardFn ProcessSegmentFn PreExpandFn PostExpandFn
ProcessExportableFn))
)

(/DECLAREDATATYPE 'ncdraft-state '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---

' 18)

(DEFINEQ

(ncdraft-do-make-draft

[LAMBDA (DocWindow DocCard NoDisplayFlg) ; Edited 2-Nov-87 20:09 by Rao
(LET* ((DocumentStyle (NCP.CardUserDataProp DocCard 'DocumentStyle))
(root-cards (NCP.CardUserDataProp DocCard 'root-cards))
(DocStream (NC.FetchSubstance DocCard))
(DocumentState)
(RESETLST
(RESETSAVE (CURSOR WAITINGCURSOR))
(NC.PrintMsg DocWindow NIL "Collecting text from descendant cards ... ")
;; Clean up the SeenBefore markers placed on the cards and boxes just copied.
[RESETSAVE NIL '(PROGN (for Card in (NCP.CardUserDataProp DocCard 'SeenCards)
do (NCP.CardUserDataProp Card 'SeenBefore NIL)

```

(NCP.CardUserDataProp Card 'SectionNumber NIL))
(NCP.CardUserDataProp DocCard 'SeenCards NIL)
(NCP.CardUserDataProp DocCard 'FixupLocations NIL)
[with ncdraft-style DocumentStyle (SETQ DocumentState
      (create ncdraft-state
        DocCard _ DocCard
        DocStream _ DocStream
        DocObj _ (TEXTOBJ DocStream)
        Section _ SectionStart
        PreCardFn _ (FUNCTION ncdraft-draft-pre-card-fn)
        ProcessSegmentFn _ (FUNCTION
          ncdraft-draft-process-segment-fn)
        PreExpandFn _ (FUNCTION ncdraft-draft-pre-expand-fn)
        PostExpandFn _ (FUNCTION ncdraft-draft-post-expand-fn)
        ProcessExportableFn _ (FUNCTION
          ncdraft-process-exportable-fn)))
  (CL:IF **NCDRAFT-COLLECT-BIB-P** (SETQ *nc-draft-bib-refs*))
  (for card in root-cards do (ncdraft-traverse-and-map-cards card DocumentState DocumentStyle))
  (for pair in (NCP.CardUserDataProp DocCard 'FixupLocations)
    do (if (SETQ LocatePtr (NCP.CardUserDataProp (CDR pair)
      'SectionNumber))
      then (TEDIT.INSERT DocStream (CONCAT SectionReferencePrefix LocatePtr)
        (CAR pair)
        NIL T)
      else (TEDIT.INSERT DocStream "***>Section-Ref<<***" (CAR pair)
        NIL T)
    (NC.PrintMsg DocWindow NIL "Done!")
    (COND
      ((NOT NoDisplayFlg)
       (BLOCK 250)
       (NC.ClearMsg DocWindow T))))))

```

(ncdraft-traverse-and-map-cards

[LAMBDA (Card DocumentState DocumentStyle) (* Rao "20-Mar-87 19:08")

(* rht 10/15/86%: Integrated markM's changes and fixed box numbering.)

```

(LET ((Type (NC.RetrieveType Card)))
  (COND
    ((OR (NCP.SketchBasedP Card)
         (NCP.GraphBasedP Card)
         (GETPROP Type 'ExportSubstanceFn))
     (NC.ActivateCardAndDo Card (ncdraft-traverse-and-map-exportable-card Card DocumentState
       DocumentStyle Type)))
    ((OR (NCP.TextBasedP Card)
         (NCP.FileBoxP Card))
     (NC.ActivateCardAndDo Card (ncdraft-traverse-and-map-text-card Card DocumentState DocumentStyle)))
    (T (NC.PrintMsg NIL NIL "Can't make document from non-exportable card " (NC.RetrieveTitle Card)
      (CHARACTER 13]))

```

(ncdraft-traverse-and-map-exportable-card

[LAMBDA (Card DocumentState DocumentStyle CardType) (* Rao "20-Mar-87 19:23")

(* Dump the CardID sketch or graph card to the document card DocStream.)

```

(DECLARE (GLOBALVARS NC.DocBackPtrLinkLabel))
(with ncdraft-state DocumentState
  (with ncdraft-style NC.DocumentStyle
    (LET ((CardStream (NC.FetchSubstance Card))
          ShrunkenFlg ThingToInsert)
      (APPLY* PreCardFn Card DocumentState DocumentStyle)
      (if (NOT (NC.FetchUserDataProp Card 'SeenBefore))
        then (SETQ ShrunkenFlg (NC.GetShrunkenWin Card))
          [NC.SetUserDataProp DocCard 'SeenCards (CONS Card (NC.FetchUserDataProp DocCard
            'SeenCards))
          (NC.SetUserDataProp Card 'SeenBefore T)
          (APPLY* ProcessExportableFn Card DocumentState DocumentStyle CardType)

```

(* Step through list of notecard imageobjs in the card we're working on and either expand or copy or ignore each according to values of ExpandEmbeddedLinks and CopyEmbeddedLinks.)

```

(for Link in (CAR (NC.CollectReferences Card NIL NIL NIL))
  bind LinkLabel ToCard ToCardType ActiveFlg ExpandP AlreadyExpanded
  everytime (BLOCK) do (SETQ LinkLabel (fetch (Link Label) of Link))
    [SETQ ExpandP (OR (EQ ExpandEmbeddedLinks 'ALL)
      (AND (LISTP ExpandEmbeddedLinks)
        (FMEMB LinkLabel ExpandEmbeddedLinks)
        (SETQ ToCard (NC.FetchUserDataProp
          (SETQ ToCard (fetch (Link DestinationCard)
            of LinkSpec))
            'SeenBefore))
          ExpandP)
    then (NC.PrintMsg NIL NIL (NC.RetrieveTitle ToCard)
      " only expanded once in this cycle."

```

```

                                (CHARACTER 13)))
                                (APPLY* PreExpandFn CardStream CurLoc DocumentState
                                DocumentStyle)
                                (if (AND ExpandFlg (NOT AlreadyExpanded))
                                then (* Expand this link. Check type and make recursive call.)
                                (ncdraft-traverse-and-map-cards ToCard DocumentState
                                DocumentStyle))
                                (APPLY* PostExpandFn CardStream CurLoc DocumentState
                                DocumentStyle)
                                finally (TEDIT.SETSEL DocStream (ADD1 (fetch TEXTLEN of DocObj))
                                0
                                'RIGHT))
                                (NC.SetUserDataProp Card 'SeenBefore NIL)
                                (AND ShrunkFlg (SHRINKW (NC.FetchWindow Card)))
                                else (NC.PrintMsg NIL NIL (NC.RetrieveTitle Card)
                                " only expanded once in this cycle."
                                (CHARACTER 13))

```

(ncdraft-traverse-and-map-text-card

[LAMBDA (Card DocumentState DocumentStyle)

; Edited 13-Aug-87 14:06 by Rao

(* * Traverse a Card Tree mapping)

(with ncdraft-state DocumentState

(with ncdraft-style DocumentStyle

(LET* ((CardStream (NC.FetchSubstance Card))

(CardObj (TEXTOBJ CardStream))

ShrunkFlg)

(APPLY* PreCardFn Card DocumentState DocumentStyle)

(if (NOT (NC.FetchUserDataProp Card 'SeenBefore))

then (SETQ ShrunkFlg (NC.GetShrunkWin Card))

[NC.SetUserDataProp DocCard 'SeenCards (CONS Card (NC.FetchUserDataProp
DocCard
'SeenCards]

(NC.SetUserDataProp Card 'SeenBefore T)

(* * Step through list of notecard imageobjs in the card we're working on and expand)

(for Object in (TEDIT.LIST.OF.OBJECTS CardObj (FUNCTION NC.LinkIconImageObjP))

bind LinkSpec LinkLabel ToCard ToCardType (LastLoc _ 1)

(CurLoc _ 0)

AlreadyExpanded ExpandP

eachtime (BLOCK)

do ((SETQ LinkSpec (NC.FetchLinkFromLinkIcon (CAR Object)))

(SETQ LinkLabel (fetch (Link Label) of LinkSpec))

(SETQ CurLoc (CADR Object)) (* Copy over any text between this obj and the last.)

(if (ILESSP LastLoc CurLoc)

then (APPLY* ProcessSegmentFn CardStream LastLoc CurLoc DocumentState
DocumentStyle))

(SETQ LastLoc (ADD1 CurLoc))

[SETQ ExpandP (OR (EQ ExpandEmbeddedLinks 'ALL)

(AND (LISTP ExpandEmbeddedLinks)

(FMEMB LinkLabel ExpandEmbeddedLinks]

(if (AND (SETQ AlreadyExpanded (NC.FetchUserDataProp (SETQ ToCard

(fetch (Link
DestinationCard
)
of LinkSpec))

'SeenBefore))

ExpandP)

then (NC.PrintMsg NIL NIL (NC.RetrieveTitle ToCard)

" only expanded once in this cycle."

(CHARACTER 13)))

(APPLY* PreExpandFn CardStream CurLoc DocumentState DocumentStyle)

(if (AND ExpandP (NOT AlreadyExpanded))

then (* Expand this link. Check type and make recursive call.)

(ncdraft-traverse-and-map-cards ToCard DocumentState DocumentStyle)

)

(APPLY* PostExpandFn CardStream CurLoc DocumentState DocumentStyle))

finally (if (ILESSP CurLoc (fetch TEXTLEN of CardObj))

then (APPLY* ProcessSegmentFn CardStream LastLoc

(ADD1 (fetch TEXTLEN of CardObj))

DocumentState DocumentStyle))

(TEDIT.SETSEL DocStream (ADD1 (fetch TEXTLEN of DocObj))

0

'RIGHT))

(NC.SetUserDataProp Card 'SeenBefore NIL)

(AND ShrunkFlg (SHRINKW (NC.FetchWindow Card)))

else (NC.PrintMsg NIL NIL (NC.RetrieveTitle Card)

" only expanded once in this cycle."

(CHARACTER 13))

)

(DEFINEQ

(ncdraft-edit-draft-style

```
[LAMBDA (window) ; (* Rao "20-Mar-87 19:09")
  (LET ((card (NCP.CardFromWindow window)))
    (ncdraft-do-edit-draft-style window (NCP.CardUserDataProp card 'DocumentStyle]))
```

(ncdraft-edit-draft-style-with-inspector

```
[LAMBDA (window) ; (* Rao "19-Mar-87 19:53")
  (INSPECT (NCP.CardUserDataProp (NCP.CardFromWindow window)
    'DocumentStyle))
```

(ncdraft-recompute-draft

```
[LAMBDA (window) ; (* Rao "20-Mar-87 19:08")
  (LET* ((card (NCP.CardFromWindow window))
    (stream (NC.FetchSubstance card)))
    [TEDIT.DELETE stream 0 (ADD1 (fetch TEXTLEN of (TEXTOBJ stream)
    (ncdraft-do-make-draft window card))])
```

)

:: Draft Card Specific Methods

(DEFINEQ

(ncdraft-draft-pre-card-fn

```
[LAMBDA (Card DocumentState DocumentStyle) ; (* Rao "20-Mar-87 19:23")
  (with ncdraft-state DocumentState (with ncdraft-style DocumentStyle
    (if (OR (EQ BuildBackLinks 'ALL)
      (AND (NEQ BuildBackLinks 'NONE)
        (FMEMB (NCP.CardType Card)
          BuildBackLinks))))
    then (NCP.LocalGlobalLink NC.DocBackPtrLinkLabel DocCard Card
      'END
      'Icon]))
```

(ncdraft-draft-process-segment-fn

```
[LAMBDA (CardStream LastLoc CurLoc DocumentState DocumentStyle) ; Edited 13-Aug-87 13:59 by Rao
  (with ncdraft-style DocumentStyle (if ProcessSubstanceP
    then (with ncdraft-state DocumentState
      (TEDIT.COPY (TEDIT.SETSEL CardStream LastLoc (IDIFFERENCE
        CurLoc LastLoc)
      )
      (TEDIT.SETSEL DocStream (fetch TEXTLEN of DocObj)
        1
        'RIGHT]))
```

(ncdraft-draft-pre-expand-fn

```
[LAMBDA (CardStream CurLoc DocumentState DocumentStyle) ; Edited 13-Dec-87 22:32 by Rao
  (with ncdraft-state DocumentState
    (with ncdraft-style DocumentStyle
      (LET (SectionP LocateP LocatePtr CopyP BibP)
        [SETQ CopyP (OR (EQ CopyEmbeddedLinks 'ALL)
          (AND (LISTP CopyEmbeddedLinks)
            (FMEMB LinkLabel CopyEmbeddedLinks))]
        [SETQ TitleP (OR (EQ TitleEmbeddedLinks 'ALL)
          (AND (LISTP TitleEmbeddedLinks)
            (FMEMB LinkLabel TitleEmbeddedLinks))]
        [SETQ SectionP (OR (EQ NumberEmbeddedLinks 'ALL)
          (AND (LISTP NumberEmbeddedLinks)
            (FMEMB LinkLabel NumberEmbeddedLinks))]
        [SETQ LocateP (OR (EQ LocateEmbeddedLinks 'ALL)
          (AND (LISTP LocateEmbeddedLinks)
            (FMEMB LinkLabel LocateEmbeddedLinks))]
        [SETQ BibP (OR (EQ BibRefEmbeddedLinks 'ALL)
          (AND (LISTP BibRefEmbeddedLinks)
            (FMEMB LinkLabel BibRefEmbeddedLinks))]
        [if (OR CopyP (AND ExpandP AlreadyExpanded))
          then ; Copy this link.
            (TEDIT.COPY (TEDIT.SETSEL CardStream CurLoc 1)
              (TEDIT.SETSEL DocStream (ADD1 (fetch TEXTLEN of DocObj)
                0
                'RIGHT))]
          [if LocateP
            then ; Put in a section pointer object
              (CL:IF **NCDRAFT-LATEX-P**
                (ncdraft-append-string-to-stream DocStream (LATEX-SECTION-PTR (NC.RetrieveTitle
                  ToCard)))
                [if (SETQ LocatePtr (NCP.CardUserDataProp ToCard 'SectionNumber))
                  then (ncdraft-append-string-to-stream DocStream (CONCAT
                    SectionReferencePrefix
                    " " LocatePtr))
                  else (NCP.CardUserDataProp DocCard 'FixupLocations
                    (CONS (CONS (ADD1 (fetch TEXTLEN of DocObj))
```

```

                                ToCard)
                                (NC.FetchUserDataProp DocCard 'FixupLocations]])]
[if TitleP
  then (CL:IF **NCDRAFT-LATEX-P**
        (ncdraft-append-string-to-stream DocStream (LATEX-SECTION Section (
                                                    NC.RetrieveTitle
                                                    ToCard)
                                                    T))
        (PROGN (ncdraft-insure-crs DocStream InsureTitleCRs)
                (ncdraft-append-string-to-stream DocStream (NC.RetrieveTitle ToCard)
                    TitleBoldP)
                (for i from 1 to TitleCRs do (ncdraft-append-string-to-stream DocStream
                    NC.CRString))))])
[if BibP
  then (CL:IF **NCDRAFT-LATEX-P**
        [ncdraft-append-string-to-stream DocStream (CL:IF **NCDRAFT-LATEX-CITE-P**
                (LATEX-CITE (NC.RetrieveTitle
                    ToCard))
                (CONVERT-BIB-STRING
                    (CONCAT "[" (NC.RetrieveTitle
                        ToCard)
                            "]")))])
        (PROGN (ncdraft-append-string-to-stream DocStream (CONCAT "[" (NC.RetrieveTitle
                    ToCard)
                        "]")))])
        (CL:IF **NCDRAFT-COLLECT-BIB-P** (push *nc-draft-bib-refs* ToCard))]
[if SectionP
  then (CL:IF **NCDRAFT-LATEX-P**
        (ncdraft-append-string-to-stream DocStream (LATEX-SECTION Section (
                                                    NC.RetrieveTitle
                                                    ToCard)))
        (PROGN (ncdraft-insure-crs DocStream InsureSectionCRs)
                (if TabSectionP
                    then (for n in (CDR Section) do (ncdraft-append-string-to-stream
                        DocStream " ")))
                (ncdraft-append-string-to-stream DocStream (CL:FORMAT NIL "~A ~A"
                    (
                        ncdraft-section-to-string
                        Section)
                    (NC.RetrieveTitle
                        ToCard))
                    SectionBoldP)
                (for i from 1 to SectionCRs do (ncdraft-append-string-to-stream DocStream
                    NC.CRString))
                (NCP.CardUserDataProp ToCard 'SectionNumber (ncdraft-section-to-string
                    Section))))
        (SETQ Section (ncdraft-section-down-level Section])

```

(ncdraft-draft-post-expand-fn

```

[LAMBDA (CardStream CurLoc DocumentState DocumentStyle) (* Rao "20-Mar-87 19:23")
  (with ncdraft-state DocumentState (with ncdraft-style DocumentStyle (LET (SectionP)
    [SETQ SectionP
      (OR (EQ NumberEmbeddedLinks
          'ALL)
          (AND (LISTP NumberEmbeddedLinks)
              (FMEMB LinkLabel
                  NumberEmbeddedLinks)
              ]
          )
      (if SectionP
          then (SETQ Section (
              ncdraft-section-up-level
              Section])

```

(ncdraft-process-exportable-fn

```

[LAMBDA (Card DocumentState DocumentStyle CardType) (* Rao "29-Apr-87 11:45")
  (with ncdraft-state DocumentState
    (** Stick an imageobj made from the card into the document. Also might be a textstream computed by the card type's
    ExportSubstanceFn.)
    [SETQ ThingToInsert (if (NCP.GraphBasedP CardType)
        then (GRAPHEROBJ CardStream)
        elseif (NCP.SketchBasedP CardType)
        then (NCP.MakeExternalSketchCopy (OR (NC.FetchWindow Card)
            CardStream))
        elseif (LET [(ExportSubstanceFn (GETPROP CardType 'ExportSubstanceFn)
            (AND ExportSubstanceFn (APPLY* ExportSubstanceFn CardStream)
            (AND CardStream (if (IMAGEOBJP ThingToInsert)
                then (TEDIT.INSERT.OBJECT ThingToInsert DocStream)
                elseif (TEXTSTREAMP ThingToInsert)
                then (TEDIT.COPY (TEDIT.SETSEL ThingToInsert 1 (fetch TEXTLEN of (TEXTOBJ
                    ThingToInsert
                    )))
                    (TEDIT.SETSEL DocStream (fetch TEXTLEN of DocObj)

```

1
'RIGHT])

)

(DEFINEQ

ncdraft-section-down-level

[LAMBDA (Section)
(CONS 1 Section)]

(* Rao "13-Mar-87 03:24")

ncdraft-section-up-level

[LAMBDA (Section)
(CONS (ADD1 (CADR Section))
(CDDR Section))

(* Rao "13-Mar-87 03:36")

ncdraft-section-to-string

[LAMBDA (Section)
(if (NULL (CDR Section))
then (CONCAT (CAR Section))
else (CONCAT (ncdraft-section-to-string (CDR Section))
"."
(CAR Section))

(* Rao "22-Mar-87 14:15")

)

:: Latex Interface Functions.

(CL:DEFVAR ****NCDRAFT-LATEX-P**** NIL)

(CL:DEFVAR ****NCDRAFT-LATEX-CITE-P**** NIL)

(CL:DEFUN **LATEX-SECTION** (SECTION TITLE &OPTIONAL NO-NUMBER-P)

(CONCAT (CASE (LENGTH SECTION)
(1 "\section")
(2 "\subsection")
(3 "\subsubsection")
(4 "\paragraph")
(5 "\subparagraph")
(CL:ERROR))
(CL:IF NO-NUMBER-P
"{"
"{"
TITLE " \label{" TITLE "}}
"))

(CL:DEFUN **LATEX-SECTION-PTR** (TITLE)

(CONCAT "\ref{" TITLE "}"))

(CL:DEFUN **LATEX-CITE** (KEY)

(CONCAT "\cite" "{" KEY "}"))

(CL:DEFUN **CONVERT-BIB-STRING** (STRING)

(LET ((CL:POSITION (CL:POSITION #\& STRING)))
(CL:IF CL:POSITION
(CL:CONCATENATE 'STRING (CL:SUBSEQ STRING 0 CL:POSITION)
"and"
(CL:SUBSEQ STRING (+ CL:POSITION 1)))
STRING)))

:: Tedit Interface Functions. First set used. Second set not used.

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NC.DocBackPtrLinkLabel NC.SubBoxLinkLabel NC.FiledCardLinkLabel PSA.Database NC.SelectingCardMenu

NC.CRString NC.DocTitleParaLeading)

)

(CL:DEFVAR ****NCDRAFT-COLLECT-BIB-P**** NIL)

(RPAQ? ***nc-draft-bib-refs***)

(DEFINEQ

ncdraft-append-string-to-stream

[LAMBDA (Stream String BoldFlg)

(* rht%: "26-Jun-85 12:17")

(* Add the String to the end of the tedit Stream.)
(* rht 11/16/84%: Now calls TEDIT.LOOKS in any case, bold or no.)
(* rht 6/26/85%: Took out call to TEDIT.LOOKS and just stuck boldfying into call to TEDIT.INSERT.)

```
(TEDIT.INSERT Stream String (ADD1 (fetch (TEXTOBJ TEXTLEN) of (TEXTOBJ Stream)))
 [FONTCOPY (TEXTPROP Stream 'FONT)
 'FACE
 (COND
 (BoldFlg 'BRR)
 (T 'MRR]
 T])
```

(ncdraft-insure-crs

[LAMBDA (stream n) (* Rao "21-Mar-87 18:31")

(* make sure there are at least n CRs at end of stream)

```
(LET ((length (fetch (TEXTOBJ TEXTLEN) of (TEXTOBJ stream)))
 (cnt 0)
 add-crs)
 (if (NOT (ZEROP length))
 then (SETFILEPTR stream (IDIFFERENCE length n)
 (for i from 1 to n do (SETQ cnt (if (EQ (BIN stream)
 13)
 then (ADD1 cnt)
 else 0)))
 (SETQ add-crs (DIFFERENCE n cnt))
 (if (NOT (ZEROP add-crs))
 then (for i from 1 to add-crs do (ncdraft-append-string-to-stream stream NC.CRString])
 )
```

(DEFINEQ

(ncdraft-fetch-to-links-in-order

[LAMBDA (Card) (* fgh%: "17-Nov-85 18:23")

(* Return the list of To links appearing in the text of ID in the order in which they appear.)

(* fgh |11/17/85| Updated to handle card object.)

```
(for ObjectPair in (TEDIT.LIST.OF.OBJECTS (TEXTOBJ (NC.FetchSubstance Card))
 (FUNCTION NC.LinkIconImageObjP))
 collect (NC.FetchLinkFromLinkIcon (CAR ObjectPair])
```

(ncdraft-add-cr-if-needed

[LAMBDA (Stream) (* Rao "20-Mar-87 19:09")

(* Check last character of Stream. If not a CR, then add one.)

```
(LET [(Len (fetch (TEXTOBJ TEXTLEN) of (TEXTOBJ Stream))
 (if (NOT (ZEROP Len))
 then (SETFILEPTR Stream (SUB1 Len))
 (if (NEQ 13 (BIN Stream))
 then (ncdraft-append-string-to-stream Stream NC.CRString])
 )
```

(ncdraft-change-para-leading

[LAMBDA (Stream) (* rht%: "16-Sep-85 19:34")

(* Change the para leading on the text stream Stream using default value.)

```
(LET ((TextObj (TEXTOBJ Stream))
 (TEDIT.PARALOOKS TextObj (LIST 'PARALEADING NC.DocTitleParaLeading)
 (fetch (TEXTOBJ TEXTLEN) of TextObj)
 1])
```

)

(DEFINEQ

(PUT-BIB-REFS

[LAMBDA NIL ; Edited 28-Nov-87 16:48 by Rao

```
(for i in *nc-draft-bib-refs* do NIL (NCP.CreateLink (NCP.WC)
 i
 'Expander
 '(T T T])
```

)

:: User Interface to Document Style Editing

(RPAQQ NC.DocumentStyleEditSpec


```
( (ExpandLinks ExpandEmbeddedLinks ALL NONE Select)
  (BackToCards BuildBackLinks ALL NONE SelCard)
  (CopyLinks CopyEmbeddedLinks ALL NONE Select)
  (TitleLinks TitleEmbeddedLinks ALL NONE Select)
  (SectionLinks NumberEmbeddedLinks ALL NONE Select)
  (ToSectionsLinks LocateEmbeddedLinks ALL NONE Select)
  (ToBibLinks BibRefEmbeddedLinks ALL NONE Select)))
```

(DEFINEQ

(ncdraft-do-edit-draft-style

[LAMBDA (draft-window draft-style wait-for-user) (* Rao "22-Mar-87 03:49")

(* Build and display an inspector window on the parameters for making documents.)

```
(LET (inspect-window (region (WINDOWREGION draft-window)))
  (SETQ inspect-window (INSPECTW.CREATE draft-style (if wait-for-user
    then (APPEND (for i in NC.DocumentStyleEditSpec
      collect (CAR i))
      ' (--DONE--))
    else (for i in NC.DocumentStyleEditSpec
      collect (CAR i)))
    (FUNCTION ncdraft-fetch-draft-style-field)
    NIL "Use left button to change values." NIL (FUNCTION
      NC.InspectorTitleCommandFn)
    "Draft Style Sheet"
    (FUNCTION ncdraft-select-draft-style-field)
    NC.OffScreenPosition NIL))
  (ATTACHWINDOW inspect-window draft-window 'TOP 'LEFT 'LOCALCLOSE)
  (WINDOWPROP inspect-window 'NoteCardsMakeDocInspector T)
  (WINDOWPROP inspect-window 'wait-for-user wait-for-user)
  (TOTOPW inspect-window)
  (if wait-for-user
    then (for while (OPENWP inspect-window) do (BLOCK))
    (WINDOWPROP inspect-window 'CancelMakeP)
  else NIL))
```

(ncdraft-fetch-draft-style-field

[LAMBDA (obj prop) (* Rao "21-Mar-87 17:49")

```
(if (EQ prop '--DONE--)
  then '--CANCEL--
  else (LET ((FieldSpec (FASSOC prop NC.DocumentStyleEditSpec)))
    (RECORDACCESS (LIST 'ncdraft-style (CADR FieldSpec))
      obj))
```

(ncdraft-select-draft-style-field

[LAMBDA (Property ValueFlg InspectWin) (* Rao "21-Mar-87 17:58")

(* Called when user buttons in the make document inspector menu on the Property parameter.
Put up a menu of choices for new values for this parameter.)

```
(if (EQ Property '--DONE--)
  then (DETACHWINDOW InspectWin)
  (CLOSEW InspectWin)
  (if ValueFlg
    then (WINDOWPROP InspectWin 'CancelMakeP ValueFlg))
  NIL
  else (LET* ((DocumentStyle (WINDOWPROP InspectWin 'DATUM))
    (OldVal (ncdraft-fetch-draft-style-field DocumentStyle Property))
    (field-spec (FASSOC Property NC.DocumentStyleEditSpec))
    (Answer (MENU (create MENU
      ITEMS _ (CDDR field-spec)
      TITLE _ "Choose New Value"))
    Links CardTypes ChangedFlg)
    [SETQ ChangedFlg (if (EQ Answer 'Select)
      then (LET ((CardWin (MAINWINDOW InspectWin)))
        (SETQ Links (NC.AskLinkLabel CardWin T T NIL NIL NIL
          (COND
            ((LISTP OldVal))
            ((EQ OldVal 'ALL)
              (NC.RetrieveLinkLabels (fetch (Card
                NoteFile
                )
              of (
                NC.CoerceToCard
                CardWin))
              T))
            (T NIL))
            T)))
        (if Links
          then (SETQ Answer (if (CAR Links)
            else 'NONE))
          (NOT (EQUAL Answer OldVal)))
        elseif (EQ Answer 'SelCard)
          then (LET ((CardWin (MAINWINDOW InspectWin)))
```

```

                (SETQ CardTypes (ncdraft-ask-card-type CardWin T NIL
                                (COND
                                  ((LISTP OldVal))
                                  ((EQ OldVal 'ALL)
                                   (NCP.CardTypes))
                                  (T NIL))
                                T)))
                (if CardTypes
                  then (SETQ Answer (if (CAR CardTypes)
                                       else 'NONE))
                    (NOT (EQUAL Answer OldVal)))
                elseif (EQ Answer 'TypeIn)
                  then (SETQ Answer (NCP.AskUser (CONCAT "Type in Value for " Property ": ")))
                    (NOT (EQUAL Answer OldVal))
                else (AND Answer (NOT (EQUAL Answer OldVal))
                (if ChangedFlg
                  then (RECORDACCESS (LIST 'ncdraft-style (CADR field-spec))
                                     DocumentStyle NIL 'REPLACE Answer)
                    (INSPECTW.REDISPLAY InspectWin Property))
                (INSPECTW.SELECTITEM InspectWin)
                NIL])

```

(ncdraft-ask-card-type

```

[LAMBDA (MainWindow MultipleFlg CancelOkayFlg OldCardTypes ReturnListOfListFlg)
  (* Rao "12-Mar-87 03:57")
  (PROG (Menu Choice Choices LabelsList CardTypes Position Card NoteFile)
    (SETQ Card (NC.CoerceToCard MainWindow))
    (SETQ NoteFile (fetch (Card NoteFile) of Card))
    (SETQ CardTypes (NCP.CardTypes))
    [SETQ Position (AND (WINDOWP MainWindow)
                       (create POSITION
                                XCOORD _ (fetch (REGION LEFT) of (WINDOWPROP MainWindow 'REGION))
                                YCOORD _ (fetch (REGION TOP) of (WINDOWREGION MainWindow))
                                (COND
                                  (MultipleFlg [SETQ Choices (STYLESHEET (CREATE.STYLE 'ITEMS (LIST (create MENU
                                                                                               'NEED.NOT.FILL.IN
                                                                                               'MULTI
                                                                                               'POSITION Position 'TITLE "Card Types?" 'SELECTIONS
                                                                                               (LIST OldCardTypes]
                                                                                               (RETURN (COND
                                                                                                 ((NULL Choices)
                                                                                                 NIL)
                                                                                                 (ReturnListOfListFlg Choices)
                                                                                                 (T (CAR Choices)
                                                                                               (SETQ Menu (create MENU
                                                                                               TITLE _ "Card Type?"
                                                                                               ITEMS _ [NCONC (COPY CardTypes)
                                                                                               (AND CancelOkayFlg (LIST '**CANCEL**')
                                                                                               MENUPOSITION _ Position))
                                                                                               (** Allow user to cancel by selecting outside of Links menu)
                                                                                               (SETQ Choice (OR (MKATOM (MENU Menu))
                                                                                               '**CANCEL**))
                                                                                               (COND
                                                                                                 ((EQ Choice '**CANCEL**')
                                                                                                 (SETQ Choice))
                                                                                               (RETURN Choice])

```

;; Register Card Type

```
(FILESLOAD NCTEXTCARD)
```

```
(DEFINEQ
```

(ncdraft-make-draft

```
[LAMBDA (Card Title NoDisplayFlg CardIdentifier) ; Edited 8-Sep-87 16:44 by Rao
```

;;; Called from a filebox's title bar. Makes a document by smashing all the descendant cards's text together.

```

(PROG (root-cards RootTitle DocWindow DocCard DocWindowOrCard DocumentStyle InspectWin)
  (OR NoDisplayFlg (SPAWN.MOUSE))
  (SETQ DocWindowOrCard (NC.ApplySupersFn MakeFn Card "Draft" NoDisplayFlg))
  ; NC.ApplySupersFn either returns a Card or a window
  ; depending on NoDisplayFlg.
  (if NoDisplayFlg
    then (SETQ DocWindow NIL)
         (SETQ DocCard DocWindowOrCard)
    else (SETQ DocWindow DocWindowOrCard)
         (SETQ DocCard (NC.CoerceToCard DocWindow)))
  ; Worry about the root card and Title.
  ; (NC.CoerceToCard CardIdentifier) to use it instead of selecting
  ; root-cards

```

```
(SETQ root-cards (NCP.SelectCards DocWindow NIL NIL "Please shift-select the Note Card or File Box the
document should start from.))
[if (NOT root-cards)
  then (NC.DeleteNoteCards DocCard T NIL NIL T T)
  (RETURN NIL)
  else (NCP.CardUserDataProp DocCard 'root-cards root-cards)
  (SETQ RootTitle (NC.RetrieveTitle (CAR root-cards)))
  (NC.SetTitle DocCard (CONCAT "Draft from %" RootTitle "%"))
  (AND DocWindow (WINDOWPROP DocWindow 'TITLE (NC.RetrieveTitle DocCard])
```

;;; Get MakeDocument parameters from user via inspector window.

```
(SETQ DocumentStyle (create ncdraft-style with *ncdraft-default-style*))
(NCP.CardUserDataProp DocCard 'DocumentStyle DocumentStyle)
[if (NOT NoDisplayFlg)
  then (if (ncdraft-do-edit-draft-style DocWindow DocumentStyle T)
    then (NC.DeleteNoteCards DocCard T NIL NIL T T)
    (RETURN NIL))
```

;;; Do it now

```
(ncdraft-do-make-draft DocWindow DocCard NoDisplayFlg)
```

;;; Exit cleanup

```
(RETURN DocWindowOrCard])
```

(ncdraft-get-draft

```
[LAMBDA (card length stream version-num) (* Rao "25-Mar-87 06:43")
  (LET ((document-style (create ncdraft-style with *ncdraft-default-style*))
    (notefile (NCP.CardNoteFile card))
    (start-loc (GETFILEPTR Stream))
    num-roots root-cards)
    (ncdraft-props-to-style (READ stream NC.OrigReadTable)
      document-style)
    (NCP.CardUserDataProp card 'DocumentStyle document-style)
    (BIN stream)
    (SETQ num-roots (NC.ReadPtr stream 2))
    (SETQ root-cards (bind card uid for i from 1 to num-roots
      when [PROGN (SETQ uid (NC.ReadUID stream))
        (AND (type? UID uid)
          (NCP.ValidCardP (SETQ card (NC.CardFromUID uid notefile)
            collect card))
        (if root-cards
          then (NCP.CardUserDataProp card 'root-cards root-cards))
        (SETQ length (DIFFERENCE length (DIFFERENCE (GETFILEPTR stream)
          start-loc)))
        (NCP.ApplySuperTypeFn GetFn card length stream version-num])
```

(ncdraft-put-draft

```
[LAMBDA (card stream) (* Rao "25-Mar-87 06:36")
  (LET* ((DocumentStyle (NCP.CardUserDataProp card 'DocumentStyle))
    (root-cards (NCP.CardUserDataProp card 'root-cards))
    (num-roots (LENGTH root-cards)))
    (PRINT (ncdraft-style-to-props DocumentStyle)
      stream NC.OrigReadTable)
    (NC.WritePtr stream num-roots 2)
    (for root in root-cards do (NC.WriteUID stream (fetch (Card UID) of root)))
    (NCP.ApplySuperTypeFn PutFn card stream])
```

)

(DEFINEQ

(NC.AddDraftCard

```
[LAMBDA NIL ; Edited 13-Aug-87 14:03 by Rao
  (NC.AddCardType 'Draft 'Text `(MakeFn %, (FUNCTION ncdraft-make-draft))
    (PutFn , (FUNCTION ncdraft-put-draft))
    (GetFn , (FUNCTION ncdraft-get-draft]
  `((LinkDisplayMode Icon)
    (DefaultHeight 500)
    (DefaultWidth 500)
    (DisplayedInMenuFlg %, T)
    (LeftButtonMenuItems %, (APPEND (NC.GetCardTypeField LeftButtonMenuItems 'Text)
      `(("Recompute Draft" (FUNCTION ncdraft-recompute-draft)
        "Recomputes this draft using current style.")
        ("Edit Draft Style" (FUNCTION ncdraft-edit-draft-style)
        "Edit the draft style of this card."
        (SUBITEMS ("Edit with Inspector" (FUNCTION
          ncdraft-edit-draft-style-with-inspector
            )
          "Edit the draft style using a Lisp
          Inspector"])
```

)

(NC.AddDraftCard)

(DEFINEQ

(NCAddStub.DraftCard

[LAMBDA NIL

; Edited 6-Aug-87 15:01 by Rao

(NC.AddCardTypeStub 'Draft 'Text 'NCDRAFTCARD NIL '((DisplayedInMenuFlg . T)

' (LinkIconAttachedBitMap])

)

::

(DECLARE%: DONTCOPY

(PUTPROPS **NCDRAFTCARD MAKEFILE-ENVIRONMENT** (:PACKAGE "IL" :READTABLE "INTERLISP"))

(PUTPROPS **NCDRAFTCARD FILETYPE** :BCOMPL)

)

(PUTPROPS **NCDRAFTCARD COPYRIGHT** ("Xerox Corporation" 1987))

FUNCTION INDEX

CONVERT-BIB-STRING	7	ncdraft-fetch-draft-style-field	9
LATEX-CITE	7	ncdraft-fetch-to-links-in-order	8
LATEX-SECTION	7	ncdraft-get-draft	11
LATEX-SECTION-PTR	7	ncdraft-insure-crs	8
NC.AddDraftCard	11	ncdraft-make-draft	10
NCAddStub.DraftCard	12	ncdraft-process-exportable-fn	6
ncdraft-add-cr-if-needed	8	ncdraft-props-to-style	2
ncdraft-append-string-to-stream	7	ncdraft-put-draft	11
ncdraft-ask-card-type	10	ncdraft-recompute-draft	5
ncdraft-change-para-leading	8	ncdraft-section-down-level	7
ncdraft-do-edit-draft-style	9	ncdraft-section-to-string	7
ncdraft-do-make-draft	2	ncdraft-section-up-level	7
ncdraft-draft-post-expand-fn	6	ncdraft-select-draft-style-field	9
ncdraft-draft-pre-card-fn	5	ncdraft-style-to-props	2
ncdraft-draft-pre-expand-fn	5	ncdraft-traverse-and-map-cards	3
ncdraft-draft-process-segment-fn	5	ncdraft-traverse-and-map-exportable-card	3
ncdraft-edit-draft-style	5	ncdraft-traverse-and-map-text-card	4
ncdraft-edit-draft-style-with-inspector	5	PUT-BIB-REFS	8

VARIABLE INDEX

NCDRAFT-COLLECT-BIB-P	7	*ncdraft-default-props*	2	NC.DocTitleParaLeading	2
NCDRAFT-LATEX-CITE-P	7	*ncdraft-default-style*	2	NC.DocumentStyleEditSpec	8
NCDRAFT-LATEX-P	7	*ncdraft-user-props*	2	NCDRAFTCARDCOMS	1
nc-draft-bib-refs	7	NC.CRString	2		

RECORD INDEX

ncdraft-state	2	ncdraft-style	2
---------------------	---	---------------------	---

PROPERTY INDEX

NCDRAFTCARD	12
-------------------	----
