

File created: 26-Jun-90 18:42:52 {DSK}<usr>local>lde>lispcore>internal>library>NSMAIL.;3

changes to: (VARS NSMAILCOMS)
(FNS NS.POLLNEWMAIL NS.OPENMAILBOX \NSMAIL.CHECK \NSMAIL.FIX.MAILBOX.LOCATIONS NS.NEXTMESSAGE
\NSMAIL.READ.ENVELOPES INBASKET.CALL NS.RETRIEVEMESSAGE \NSMAIL.RETRIEVE \NSMAIL.SIGNAL.ERROR
NS.CLOSEMAILBOX \NSMAIL.LOGOFF \NSMAIL.CHANGE.STATUS \MAILOBJ.DISPLAY \MAILOBJ.IMAGEBOX
\MAILOBJ.PUT \MAILOBJ.BUTTONEVENTFN \MAILOBJ.HARDCOPY \MAILOBJ.FB \MAILOBJ.PUT.FILE
\MAILOBJ.VIEW \MAILOBJ.EXPAND \NSMAIL.SEND)

previous date: 14-Feb-90 17:23:04 {DSK}<usr>local>lde>lispcore>internal>library>NSMAIL.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1984, 1985, 1986, 1987, 1988, 1989, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ NSMAILCOMS

```
[ (COMS
(COURIERPROGRAMS MAILTRANSPORT INBASKET)
(FNS \NSMAIL.AUTHENTICATE \NSMAIL.MAKE.MAILSERVERS \NSMAIL.LOGIN NS.FINDMAILBOXES)
(ALISTS (LAFITEMODELST NS STAR)))
; Basic mail protocol

(COMS
(FNS NS.POLLNEWMAIL NS.OPENMAILBOX \NSMAIL.CHECK \NSMAIL.FIX.MAILBOX.LOCATIONS NS.NEXTMESSAGE
 \NSMAIL.READ.ENVELOPES INBASKET.CALL NS.RETRIEVEMESSAGE \NSMAIL.RETRIEVE
 \NSMAIL.EOF.ON.RETRIEVE \NSMAIL.READ.SERIALIZED.TREE \NSMAIL.CHECK.SERIALIZED.VERSION
 \NSMAIL.READ.SERIALIZED.CONTENT \NSMAIL.DISCARD.SERIALIZED.CONTENT
 \NSMAIL.READ.STRING.AS.STREAM \NSMAIL.PRINT.HEADERFIELDS \NSMAIL.PRINT.NAMES)
; Retrieving mail

(FNS \NSMAIL.COURIER.OPEN \NSMAIL.ERRORHANDLER \NSMAIL.SIGNAL.ERROR)
; Error handling

(FNS NS.CLOSEMAILBOX \NSMAIL.LOGOFF \NSMAIL.CHANGE.STATUS)
; Close/flush protocol

[INITVARS (NSMAILDEBUGFLG)
(NSMAIL.LEAVE.ATTACHMENTS)
(NSMAIL.HEADER.ORDER ' (Date Sender From Subject In-Reply-to To cc Message-ID Reply-to)
(ADDVARS (\NSMAIL.GOOD.BODYTYPES 2 4)))
; Handling attachments as a special kind of image object

[COMS
(FNS \MAILOBJ.CREATE \MAILOBJ.TYPE.NAME \MAILOBJ.NS.TO.LISP.NAME \MAILOBJ.DISPLAY \MAILOBJ.GET
 \MAILOBJ.IMAGEBOX \MAILOBJ.PUT \MAILOBJ.INIT)
(FNS \MAILOBJ.BUTTONEVENTFN \MAILOBJ.DO.COMMAND \MAILOBJ.HARDCOPY \MAILOBJ.FB \MAILOBJ.PUT.FILE
 \MAILOBJ.VIEW \MAILOBJ.MUNGE.NAME \MAILOBJ.COPY.BODY \MAILOBJ.EXPAND \MAILOBJ.COPY.CHILD
 \MAILOBJ.COPY.SEQUENCE \MAILOBJ.EXTRACT.TEXT \MAILOBJ.PARSE.ATTRIBUTES)
(ADDVARS (FILING.TYPES (VIEWPOINT 4353)
 (RES 4428)
 (XEROX860 5120)
 (REFERENCE 4427)
 (MAILFOLDER 4417)))
(VARS MAILOBJ.REFERENCE.FIELD)
(INITVARS (MAILOBJ.WINDOWOFFSET 16)
 (MAILOBJ.SKIPCHAR (CHARCODE ".")))
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS MAILOBJ)
 (CONSTANTS \MAILOBJ.REFERENCE.LAST.FILED))
(DECLARE%: DONTEVAL@LOAD DOCOPY (P (\MAILOBJ.INIT
 (AND (EQ MAKESYSNAME :LYRIC)
 (FILESLOAD (SYSLOAD)
 NSRANDOM]
; sending mail

(COMS
(FNS \NSMAIL.SEND.PARSE \NSMAIL.PARSE.REFERENCE \NSMAIL.EXPAND.DL \NSMAIL.PARSE \NSMAIL.PARSE1
 NS.REMOVEDUPLICATES \NSMAIL.SEND \NSMAIL.PREPARE.ATTACHMENT \NSMAIL.GUESS.FILE.TYPE
 \NSMAIL.SEND.MESSAGE.CONTENT COURIER.WRITE.STREAM.UNSPECIFIED \NSMAIL.SEND.STREAM.AS.STRING
 \NSMAIL.WRITE.ATTRIBUTE \NSMAIL.FINDSERVER \NSMAIL.CHECKSERVER)
(FILES LAFITEMAIL)
; for LAFITE.MAKE.PARSE.TABLE

(VARS NSMAIL.PARSEFIELDS (\LAPARSE.NSMAIL (LAFITE.MAKE.PARSE.TABLE NSMAIL.PARSEFIELDS))
(GLOBALVARS \LAPARSE.NSMAIL)
(INITVARS (\NSMAIL.SERVER.CACHE)
(NSMAIL.NET.HINT)
(*NSMAIL-MAX-NOTE-LENGTH* 8000)
(*NSMAIL-SEND-MAIL-NOTES*)
(NSMAIL-CACHE-TIMEOUT* 14400000)
(LAFITEDL.EXT "DL"))
[P (CL:PROCLAIM ' (GLOBAL NSMAIL.NET.HINT *NSMAIL-MAX-NOTE-LENGTH* *NSMAIL-SEND-MAIL-NOTES*
 *NSMAIL-CACHE-TIMEOUT*)
(ADDVARS (\SYSTEMCACHEVARS \NSMAIL.SERVER.CACHE))
(FNS \NSMAIL.MESSAGE.P \NSMAIL.MESSAGE.FROM.SELF.P \NSMAIL.MAKEANSWERFORM))
; Utility for handling mail attributes

(COMS
(PROP COURIERDEF ENVELOPE.ITEM)
(FNS \NS.READ.ENVELOPE.ITEM \NS.WRITE.ENVELOPE.ITEM)
(VARS \NSMAIL.ENVELOPE.ITEM.TYPES)
(DECLARE%: EVAL@COMPILE DOCOPY (VARS \NSMAIL.ATTRIBUTES)))
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS NSMAILBOX NSMAILSTATE NSMAILPARSE)
 (CONSTANTS \NSMAIL.SOCKET \SERIALIZED.FILE.VERSION \SERIALIZED.FILE.VERSIONS
 \NSMAIL.CTSTANDARD.MESSAGE \NSMAIL.TEXT.BODYTYPE \NSMAIL.EMPTY.BODYTYPE
```

```

        \NSMAIL.REFERENCE.BODYTYPE MAX.BULK.SEGMENT.LENGTH \NULL.CACHE.VERIFIER)
(MACROS \NSMAIL.ATTRIBUTE.TYPE \NSMAIL.WRITE.ATTRIBUTE \NSMAIL.WRITE.ATTRIBUTE.MACRO)
(PROP INFO \NSMAIL.ATTRIBUTE.TYPE)
(GLOBALVARS NSMAIL.NET.HINT \NSMAIL.ENVELOPE.ITEM.TYPES \NSMAIL.ATTRIBUTES \NSMAIL.SERVER.CACHE
NSMAILDEBUGFLG NSWIZARDFLG NSMAIL.LEAVE.ATTACHMENTS \NSMAIL.GOOD.BODYTYPES
MAILOBJ.WINDOWOFFSET MAILOBJ.SKIPCHAR \MAILOBJ.IMAGEFNS MAILOBJ.REFERENCE.FIELD
\NSMAIL.ATTRIBUTES DEFAULTICONFONT NSPRINT.WATCHERFLG NSMAIL.HEADER.ORDER FILING.TYPES)
[P (CL:PROCLAIM ' (CL:SPECIAL *RETRIEVAL-ERROR*)
(FILE (SOURCE)
LAFITEDECLS)
(FILE (LOADCOMP)
CLEARINGHOUSE)
(LOCALVARS . T))
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS (ADDVARS (NLAMA)
(NLAML)
(LAMA INBASKET.CALL))

```

:: Basic mail protocol

(COURIERPROGRAM MAILTRANSPORT (17 4)

TYPES

```

[(CREDENTIALS (AUTHENTICATION . CREDENTIALS))
(VERIFIER (AUTHENTICATION . VERIFIER))
(ENVELOPE.ITEM.TYPE LONGCARDINAL)
(ENVELOPE (SEQUENCE ENVELOPE.ITEM))
(INVALID.NAME (RECORD (REASON INVALID.NAME.REASON)
(NAME RNAME)))
(INVALID.NAME.LIST (SEQUENCE INVALID.NAME))
(INVALID.NAME.REASON (ENUMERATION (NoSuchRecipient 0)
(CantValidateNow 1)
(IllegalName 2)
(Refused 3)
(NoAccessToDl 4)
(Timeout 5)
(NoDlsAllowed 6)
(MessageTooLong 7)))
(NAME (CLEARINGHOUSE . NAME))
(NAME.LIST (SEQUENCE NAME))
(RNAME NAME)
(RNAME.LIST (SEQUENCE RNAME))
(WILLINGNESS CARDINAL)
(CONTENTS.TYPE LONGCARDINAL)
(MESSAGEID (ARRAY 5 UNSPECIFIED))
(POSTMARK (RECORD (POSTED.AT NAME)
(TIME TIME)))
(PROBLEM (RECORD (UNDELIVERABLES INVALID.NAME.LIST)
(RETURNED.ENVELOPE ENVELOPE)))
(CONNECTION.PROBLEM (FILING . CONNECTION.PROBLEM))
(SERVICE.PROBLEM (ENUMERATION (CannotAuthenticate 0)
(ServiceFull 1)
(ServiceUnavailable 2)
(MediumFull 3)))
(TRANSFER.PROBLEM (ENUMERATION (Aborted 0)
(NoRendezvous 1)
(WrongDirection 4)

```

PROCEDURES

```

((SERVER.POLL 0 (CREDENTIALS VERIFIER)
RETURNS
(WILLINGNESS (CLEARINGHOUSE . NETWORK.ADDRESS.LIST)
VERIFIER NAME))
(POST 1 (CREDENTIALS VERIFIER RNAME.LIST BOOLEAN BOOLEAN CONTENTS.TYPE ENVELOPE BULK.DATA.SOURCE)
RETURNS
(INVALID.NAME.LIST MESSAGEID)
REPORTS
(AUTHENTICATION.ERROR CONNECTION.ERROR INVALID.RECIPIENTS SERVICE.ERROR TRANSFER.ERROR
UNDEFINED.ERROR)))

```

ERRORS

```

((AUTHENTICATION.ERROR 1 ((AUTHENTICATION . PROBLEM)))
(CONNECTION.ERROR 2 (CONNECTION.PROBLEM))
(INVALID.RECIPIENTS 3 (INVALID.NAME.LIST))
(SERVICE.ERROR 4 (SERVICE.PROBLEM))
(TRANSFER.ERROR 5 (TRANSFER.PROBLEM))
(UNDEFINED.ERROR 6 (CARDINAL)))

```

(COURIERPROGRAM INBASKET (18 1)

INHERITS

(MAILTRANSPORT)

TYPES

```

[(CREDENTIALS (AUTHENTICATION . CREDENTIALS))
(VERIFIER (AUTHENTICATION . VERIFIER))
(SESSION (RECORD (HANDLE (ARRAY 2 UNSPECIFIED))
(VERIFIER VERIFIER)))
(ENVELOPE.ITEM.TYPE LONGCARDINAL)
(ENVELOPE (SEQUENCE ENVELOPE.ITEM))
(INVALID.NAME (RECORD (REASON INVALID.NAME.REASON)
(NAME RNAME)))
(INVALID.NAME.LIST (SEQUENCE INVALID.NAME))

```

```

(INVALID.NAME.REASON (ENUMERATION (NoSuchRecipient 0)
                                (CantValidateNow 1)
                                (IllegalName 2)
                                (Refused 3)
                                (NoAccessToDl 4)
                                (Timeout 5)
                                (NoDlsAllowed 6)
                                (MessageTooLong 7)))
(NAME (CLEARINGHOUSE . NAME))
(NAME.LIST (SEQUENCE NAME))
(RNAME NAME)
(RNAME.LIST (SEQUENCE RNAME))
(CONTENTS.TYPE LONGCARDINAL)
(INDEX CARDINAL)
(INBASKET.STATE (RECORD (LASTINDEX INDEX)
                        (NEWCOUNT CARDINAL)
                        (ISPRIMARY BOOLEAN)
                        (ISPRIMARYUP BOOLEAN)))
(RANGE (RECORD (FIRST INDEX)
               (LAST INDEX)))
(MAIL.ATTRIBUTE.TYPE LONGCARDINAL)
[MAIL.ATTRIBUTE (RECORD (TYPE MAIL.ATTRIBUTE.TYPE)
                       (VALUE (SEQUENCE UNSPECIFIED)
                               (TRANSPORT.ENVELOPE BOOLEAN)
                               (INBASKET.ENVELOPE BOOLEAN)
                               (MAIL.ATTRIBUTES (SEQUENCE MAIL.ATTRIBUTE.TYPE)
                                                 (CACHE.VERIFIER (ARRAY 4 UNSPECIFIED))
                                                 (MESSAGE.DESCRPTION (RECORD (MESSAGE.INDEX INDEX)
                                                                    (TRANSPORT.ENVELOPE ENVELOPE)
                                                                    (INBASKET.ENVELOPE ENVELOPE)
                                                                    (CONTENT.ATTRIBUTES ENVELOPE))))
                               (CACHE.STATUS UNSPECIFIED)
                               (STATUS (ENUMERATION (NEW 0)
                                                    (KNOWN 1)
                                                    (RECEIVED 2))))
                               (ACCESS.PROBLEM (ENUMERATION (AccessRightsInsufficient 0)
                                                            (AccessRightsIndeterminate 1)
                                                            (InbasketInUse 2)
                                                            (NoSuchRecipients 3)
                                                            (RecipientNameIndeterminate 4)))
                               (CONNECTION.PROBLEM (FILING . CONNECTION.PROBLEM))
                               (SERVICE.PROBLEM (ENUMERATION (CannotAuthenticate 0)
                                                             (ServiceFull 1)
                                                             (ServiceUnavailable 2)
                                                             (MediumFull 3)))
                               (TRANSFER.PROBLEM (ENUMERATION (Aborted 0)
                                                                (NoRendezvous 1)
                                                                (WrongDirection 4)))
                               (SESSION.PROBLEM (ENUMERATION (TokenInvalid 0)
                                                             (SessionInUse 1)))
                               (CALL.PROBLEM (ENUMERATION (USE.COURIER 0)
                                                         (LOGON 5 (CREDENTIALS VERIFIER NAME CACHE.VERIFIER BOOLEAN)
                                                         RETURNS
                                                         (SESSION CACHE.STATUS)
                                                         REPORTS
                                                         (ACCESS.ERROR AUTHENTICATION.ERROR SERVICE.ERROR UNDEFINED.ERROR))
                                                         (LOGOFF 4 (SESSION)
                                                         RETURNS
                                                         (CACHE.VERIFIER)
                                                         REPORTS
                                                         (AUTHENTICATION.ERROR SESSION.ERROR UNDEFINED.ERROR))
                                                         (MAILPOLL 7 (CREDENTIALS VERIFIER NAME)
                                                         RETURNS
                                                         (INBASKET.STATE)
                                                         REPORTS
                                                         (ACCESS.ERROR AUTHENTICATION.ERROR SERVICE.ERROR UNDEFINED.ERROR))
                                                         (MAILCHECK 6 (SESSION)
                                                         RETURNS
                                                         (INBASKET.STATE CARDINAL)
                                                         REPORTS
                                                         (ACCESS.ERROR AUTHENTICATION.ERROR SERVICE.ERROR UNDEFINED.ERROR))
                                                         (CHANGE.STATUS 0 (SESSION RANGE STATUS)
                                                         RETURNS NIL REPORTS (AUTHENTICATION.ERROR INVALID.INDEX SESSION.ERROR UNDEFINED.ERROR))
                                                         (DELETE 1 (SESSION RANGE)
                                                         RETURNS NIL REPORTS (AUTHENTICATION.ERROR INVALID.INDEX SESSION.ERROR UNDEFINED.ERROR))
                                                         (LIST 2 (SESSION RANGE SELECTIONS BULK.DATA.SINK)
                                                         RETURNS NIL REPORTS (AUTHENTICATION.ERROR CONNECTION.ERROR INVALID.INDEX SESSION.ERROR
                                                         TRANSFER.ERROR UNDEFINED.ERROR))
                                                         (LOCATE 3 (SESSION STATUS)
                                                         RETURNS
                                                         (INDEX)
                                                         REPORTS
                                                         (AUTHENTICATION.ERROR SESSION.ERROR UNDEFINED.ERROR))
                                                         (RETRIEVE 8 (SESSION INDEX CONTENTS.TYPE BULK.DATA.SINK)
                                                         RETURNS
                                                         (ENVELOPE ENVELOPE)

```

PROCEDURES

REPORTS
(AUTHENTICATION.ERROR CONNECTION.ERROR CONTENTS.TYPE.MISMATCH INVALID.INDEX SESSION.ERROR
TRANSFER.ERROR UNDEFINED.ERROR))

ERRORS

((ACCESS.ERROR 0 (ACCESS.PROBLEM))
(AUTHENTICATION.ERROR 1 ((AUTHENTICATION . PROBLEM)))
(CONNECTION.ERROR 2 (CONNECTION.PROBLEM))
(CONTENTS.TYPE.MISMATCH 3 (CONTENTS.TYPE))
(SESSION.ERROR 5 (SESSION.PROBLEM))
(INVALID.INDEX 4 (INDEX))
(SERVICE.ERROR 6 (SERVICE.PROBLEM))
(TRANSFER.ERROR 7 (TRANSFER.PROBLEM))
(UNDEFINED.ERROR 8 (CALL.PROBLEM)))

(DEFINEQ

(\NSMAIL.AUTHENTICATE

; Edited 5-Jan-90 18:36 by bvm

[LAMBDA NIL
(LET ((INFO (\INTERNAL/GETPASSWORD ' |NS::|))
NSUSERNAME FULLNAME MSERVERS AUTHENTICATED? CREDENTIALS MSG)
(SETQ NSUSERNAME (PARSE.NSNAME (CAR INFO)))
(COND
((NEQ [SETQ AUTHENTICATED? (COND
((NULL (SETQ FULLNAME (CH.LOOKUP.OBJECT NSUSERNAME)))
'NoSuchUser)
(T (NS.AUTHENTICATE (SETQ CREDENTIALS (NS.MAKE.SIMPLE.CREDENTIALS
(CONS FULLNAME (CDR INFO)
T)
(printout PROMPTWINDOW T "Cannot authenticate user " (NSNAME.TO.STRING (OR FULLNAME NSUSERNAME)
T)
" because: "
(SELECTQ (SETQ \LAFITE.AUTHENTICATION.FAILURE AUTHENTICATED?)
(CredentialsInvalid
>Login incorrect")
(KeysUnavailable
(CONCAT "Authentication server unavailable for domain " (fetch NSDOMAIN of FULLNAME)
))
(NoSuchUser "No such user")
AUTHENTICATED?)
".")
NIL)
(T (create LAFITEMODEDATA
FULLUSERNAME _ (NSNAME.TO.STRING FULLNAME T)
UNPACKEDUSERNAME _ FULLNAME
CREDENTIALS _ CREDENTIALS
SHORTUSERNAME _ (CONCAT (fetch NSOBJECT of FULLNAME)
' %:
(COND
((NOT (STRING-EQUAL (fetch NSDOMAIN of FULLNAME)
CH.DEFAULT.DOMAIN))
(fetch NSDOMAIN of FULLNAME))
(T "")))
MAILSERVERS _ (\NSMAIL.MAKE.MAILSERVERS (NS.FINDMAILBOXES FULLNAME)
FULLNAME CREDENTIALS])

(\NSMAIL.MAKE.MAILSERVERS

; Edited 16-Aug-89 16:05 by bvm

[LAMBDA (SERVERS FULLNAME CREDENTIALS)
;; Return a list of mail server info for insertion in the MAILSERVERS slot of NS mode. Each element of SERVERS is of the form (name .
;; addresses)
(if (NULL SERVERS)
then (printout PROMPTWINDOW T "There are no mail servers for user " (NSNAME.TO.STRING FULLNAME T))
NIL
else (for PAIR in SERVERS bind (FIRSTTIME _ T) collect (create MAILSERVER
MAILPORT _ (CADR PAIR)
MAILSERVERNAME _ (CAR PAIR)
MAILSERVEROPS _
(CONSTANT (LIST (FUNCTION NS.POLLNEWMAIL)
(FUNCTION NS.OPENMAILBOX)
(FUNCTION NS.NEXTMESSAGE)
(FUNCTION NS.RETRIEVEMESSAGE)
(FUNCTION NS.CLOSEMAILBOX)))
MAILSTATE _ (create NSMAILSTATE
STATENAME _ FULLNAME
STATEADDRESS _ (CADR PAIR)
STATECREDENTIALS _
CREDENTIALS
STATETIMER _
(if FIRSTTIME
then
; Only need a timer on the first server
(SETQ FIRSTTIME NIL)
(SETUPTIMER
NSMAIL-CACHE-TIMEOUT
]))

(\NSMAIL.LOGIN

```
[LAMBDA NIL ; Edited 7-Jun-88 19:37 by bvm
  (if (LAFITE.PROMPT.FOR.LOGIN ' |NS::|)
      then ; Got the login, now authenticate
        (\LAFITE.GET.USER.DATA 'NS NIL T)
        (\LAFITE.WAKE.WATCHER])
```

(NS.FINDMAILBOXES

```
[LAMBDA (USERNAME) ; Edited 18-Jul-88 12:55 by bvm
  (LET [(MAILBOXENTRY (CH.RETRIEVE.ITEM (PARSE.NSNAME USERNAME)
    (CH.PROPERTY 'MAILBOXES)
    'MAILBOX.VALUES)
    (AND MAILBOXENTRY (for MB in (COURIER.FETCH (CLEARINGHOUSE . MAILBOX.VALUES)
      MAIL.SERVICE of (CADR MAILBOXENTRY))
      when (SETQ MB (COND
        ((LOOKUP.NS.SERVER MB NIL T))
        (T (PRINTOUT PROMPTWINDOW T "Cannot find address for mail server
" MB)
          NIL)))
      collect MB])
  )
```

```
(ADDTOVAR LAFITEMODELST (NS 1 \NSMAIL.SEND.PARSE \NSMAIL.SEND \NSMAIL.MAKEANSWERFORM \NSMAIL.AUTHENTICATE
  \NSMAIL.MESSAGE.P \NSMAIL.MESSAGE.FROM.SELF.P \NSMAIL.LOGIN)
  (STAR . NS))
```

:: Retrieving mail

(DEFINEQ

(NS.POLLNEWMAIL

```
[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER) ; Edited 26-Jun-90 18:21 by jds
  (LET (RESULT N)
    (COND
      ((NOT (SETQ RESULT (\NSMAIL.CHECK ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER)))
        ; Server down
        '?)
      ((AND (> (SETQ N (fetch (NSMAILSTATE STATEFIRSTNEW) of (fetch MAILSTATE of MAILSERVER)))
        0)
        (> (SETQ N (ADD1 (- (COURIER.FETCH (INBASKET . INBASKET.STATE)
          LASTINDEX of RESULT)
          N)))
        ; Return number of messages
        0))
      N]))
```

(NS.OPENMAILBOX

```
[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER) ; Edited 26-Jun-90 18:21 by jds
  (LET ((STREAM (\NSMAIL.COURIER.OPEN ADDRESS)
    NSMAILSTATE INBASKETSTATE FIRSTINDEX LASTINDEX N)
    (COND
      ((NULL STREAM)
        NIL)
      ((OR (NULL (SETQ INBASKETSTATE (\NSMAIL.CHECK ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER STREAM T
        )))
        (EQ (CAR INBASKETSTATE)
          'ERROR))
        ; Return error msg
        (CLOSEF STREAM)
        (CONS NIL (CDR INBASKETSTATE)))
      ((EQ [SETQ N (COND
        ((EQ [SETQ FIRSTINDEX (fetch (NSMAILSTATE STATEFIRSTNEW)
          of (SETQ NSMAILSTATE (fetch MAILSTATE of MAILSERVER]
          0)
          ; No NEW messages at all
          (T
            ; Protocol suggests using (courier.fetch (inbasket .
            ; inbasket.state) newcount inbasketstate) but that's always zero.
            (ADD1 (- (SETQ LASTINDEX (COURIER.FETCH (INBASKET . INBASKET.STATE)
              LASTINDEX of INBASKETSTATE))
              FIRSTINDEX]
            0)
            (\NSMAIL.LOGOFF NSMAILSTATE STREAM)
            'EMPTY)
            (T
              ; Return (MAILBOX . properties)
              (CONS (create NSMAILBOX
                NSMAILSTREAM _ STREAM
                NSMAILLASTINDEX _ LASTINDEX
                NSMAILSTATE _ NSMAILSTATE)
                (LIST '%OFMESSAGES N]))
```

(\NSMAIL.CHECK

```
[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER STREAM RETURNERRORS)
  ; Edited 26-Jun-90 18:21 by jds
```

;;; Performs a mail check for user REGISTEREDNAME at ADDRESS, returning INBASKETSTATE if successful, NIL if not. Updates the MAILSTATE of
;;; MAILSERVER as appropriate to reflect current SESSION and STATEFIRSTNEW (first new message)

```

(RESETLST
  (PROG ((JUSTCHECKING (NULL STREAM))
        (STATE (fetch (MAILSERVER MAILSTATE) of MAILSERVER))
        SESSION POLLRESULT LASTINDEX FIRSTNEW OLDLAST CONTINUANCE TIMER)
    (COND
      ((AND JUSTCHECKING (SETQ TIMER (fetch (NSMAILSTATE STATETIMER) of STATE))
        (TIMEREXPIRED? TIMER)
        (\NSMAIL.FIX.MAILBOX.LOCATIONS))
        ; Some mailboxes moved
        (GO FAILFAST)))
      (SETQ SESSION (fetch (NSMAILSTATE STATESESSION) of STATE))
      (SETQ FIRSTNEW (fetch (NSMAILSTATE STATEFIRSTNEW) of STATE))
      (SETQ OLDLAST (fetch (NSMAILSTATE STATEOLDLAST) of STATE))
    )
  )
  RETRY
  [COND
    ((NULL SESSION)
     (if (AND (NOT NSMAIL.LEAVE.ATTACHMENTS)
              JUSTCHECKING)
          then
            ; Just polling, don't need session
            (SETQ POLLRESULT (COURIER.EXPEDITED.CALL ADDRESS \NSMAIL.SOCKET 'INBASKET
              'MAILPOLL
              (CAR CREDENTIALS)
              (CDR CREDENTIALS)
              (fetch (NSMAILSTATE STATENAME) of STATE)
              'RETURNERRORS))
            (GO GOTRESULT))
          )
    )
    [COND
      ((NULL STREAM)
       ; Need a real Courier stream for some reason here
       (COND
         ((SETQ STREAM (COURIER.OPEN ADDRESS NIL T 'NSMAIL))
          (RESETSAVE NIL (LIST 'CLOSEF STREAM)))
          (T (RETURN NIL))
        )
      )
      (COND
        ((EQ [CAR (SETQ SESSION (COND
          ((OR T STREAM) ; Would be nice to do this expedited, but this ability was taken
          ; out in Services 8.1!
          (COURIER.CALL STREAM 'INBASKET 'LOGON (CAR CREDENTIALS)
            (CDR CREDENTIALS)
            (fetch (NSMAILSTATE STATENAME) of STATE)
            \NULL.CACHE.VERIFIER T 'RETURNERRORS))
          (T (COURIER.EXPEDITED.CALL ADDRESS \NSMAIL.SOCKET 'INBASKET
            'LOGON
            (CAR CREDENTIALS)
            (CDR CREDENTIALS)
            (fetch (NSMAILSTATE STATENAME) of STATE)
            \NULL.CACHE.VERIFIER T 'RETURNERRORS])
          'ERROR)
          (GO ERROR)))
         (replace (NSMAILSTATE STATESESSION) of STATE with (SETQ SESSION (CAR SESSION))
        )
      ]
      [SETQ POLLRESULT (COND
        ((NULL STREAM) ; Just checking
         (COURIER.EXPEDITED.CALL ADDRESS \NSMAIL.SOCKET 'INBASKET 'MAILCHECK SESSION
           'RETURNERRORS))
         (T (COURIER.CALL STREAM 'INBASKET 'MAILCHECK SESSION 'RETURNERRORS])
        )
      )
    ]
  )
  GOTRESULT
  [COND
    ((NULL POLLRESULT) ; Failed somehow
     (RETURN NIL))
    ((EQ (CAR (LISTP POLLRESULT))
         'ERROR)
     (COND
       ((EQ (CADR POLLRESULT)
            'SESSION.ERROR)
        ; Session timed out, start a new one
        (replace (NSMAILSTATE STATESESSION) of STATE with (SETQ SESSION NIL))
        (replace (NSMAILSTATE STATEFIRSTNEW) of STATE with (SETQ FIRSTNEW NIL))
        (replace (NSMAILSTATE STATEOLDLAST) of STATE with (SETQ OLDLAST NIL))
        (GO RETRY))
       (T (SETQ SESSION POLLRESULT)
          (GO ERROR])
      )
    )
    (replace (NSMAILSTATE STATELASTERROR) of STATE with NIL)
    (if SESSION
     then
       ; MAILCHECK returned 2 values: state and continuance
       (SETQ CONTINUANCE (CADR POLLRESULT))
       (SETQ POLLRESULT (CAR POLLRESULT)))
    )
    (COND
      ((EQ (SETQ LASTINDEX (COURIER.FETCH (INBASKET . INBASKET.STATE)
        LASTINDEX of POLLRESULT))
          0)
        ; Mailbox is empty
        (replace (NSMAILSTATE STATEFIRSTNEW) of STATE with 0))
      (NOT NSMAIL.LEAVE.ATTACHMENTS) ; Retrieving all mail, so we don't care about NEW vs OLD
      (replace (NSMAILSTATE STATEFIRSTNEW) of STATE with 1)
      (replace (NSMAILSTATE STATEOLDLAST) of STATE with LASTINDEX))
      ((OR (NULL OLDLAST)
          (ILESSP OLDLAST LASTINDEX)
          (NOT JUSTCHECKING)
        )
      )
    )
  )

```

```

                (NULL FIRSTNEW)) ; Need to accurately locate first NEW message
[replace (NSMAILSTATE STATEFIRSTNEW) of STATE with (COND
                (STREAM (COURIER.CALL STREAM 'INBASKET
                        'LOCATE SESSION
                        'NEW
                        'NOERROR))
                (T (COURIER.EXPEDITED.CALL ADDRESS
                    \NSMAIL.SOCKET 'INBASKET
                    'LOCATE SESSION 'NEW
                    'RETURNERRORS])

                (replace (NSMAILSTATE STATEOLDLAST) of STATE with LASTINDEX)))
[replace (MAILSERVER CONTINUANCE) of MAILSERVER with (AND (FIXP CONTINUANCE)
                (ITIMES 1000 (IQUOTIENT (ITIMES
                CONTINUANCE
                4)
                5]
                ; Tell poller to call again soon enough to keep session alive

(RETURN POLLRESULT)
ERROR
[if [AND [NOT (EQUAL (CDR SESSION)
                    ' (CONNECTION.PROBLEM NoResponse]
    (NOT (EQUAL (CDR SESSION)
                (fetch (NSMAILSTATE STATELASTERROR) of STATE])
then
;; Don't bother mentioning the error if it's just a timeout, since mailwatch will handle our NIL response fine. Also don't
;; repeatedly print the same error message.
(replace (NSMAILSTATE STATELASTERROR) of STATE with (CDR SESSION))
(LET [(ERRMSG (CASE (CADR SESSION)
                    (REJECT) ; 3rd element = (reason ...)
                    (CAADDR SESSION)
                    ((SERVICE.ERROR ACCESS.ERROR)
                    ; the specific reason is just as informative, and more readable
                    ; than the whole error.
                    (CADDR SESSION))
    (T (COND
        (NSWIZARDFLG (HELP SESSION)))
        (SUBSTRING (CDR SESSION)
                    2 -2)))]
(if RETURNERRORS
then (RETURN (CONS 'ERROR ERRMSG))
elseif (AND (EQ ERRMSG 'NoSuchRecipients)
            (\NSMAIL.FIX.MAILBOX.LOCATIONS))
then
;; Rather odd message. We get this when the server no longer holds this inbox. At this point we have
;; fixed mail servers in NS mode, but there's no good way for us to report the news, so go ahead and
;; return NIL, but set "continuance" so that poll will happen again immediately
(replace (MAILSERVER CONTINUANCE) of MAILSERVER with 0)
else (LET ((*PRINT-CASE* :UPCASE)) ; Lousy atomic error names...
        (CL:FORMAT PROMPTWINDOW "~%From mail server ~A: ~A" (fetch (MAILSERVER
                MAILSERVERNAME
                )
                of MAILSERVER)
        (CASE ERRMSG
            (NoSuchService "Mail service not running")
            (T ERRMSG)))]
(RETURN NIL)
FAILFAST)))]

```

(\NSMAIL.FIX.MAILBOX.LOCATIONS

```

[LAMBDA NIL
                ; Edited 26-Jun-90 18:21 by jds
;; Called when we think user's mailboxes may have moved. If they have, sets new info into NS mode and returns T.
(LET ((OLDDATA (\LAFITE.GET.USER.DATA 'NS))
      OLDSERVERS NEWSERVERS FULLNAME)
  (if (AND OLDDATA (SETQ OLDSERVERS (fetch (LAFITEMODEDATA MAILSERVERS) of OLDDATA)))
      then
      [SETQ NEWSERVERS (NS.FINDMAILBOXES (SETQ FULLNAME (fetch (LAFITEMODEDATA UNPACKEDUSERNAME)
                                                              of OLDDATA))
      [LET [(STATE (fetch (MAILSERVER MAILSTATE) of (CAR OLDSERVERS))
              ; Reset the timer that tells us when next to check on location.
              (replace (NSMAILSTATE STATETIMER) of STATE with (SETUPTIMER (if NEWSERVERS
              then *NSMAIL-CACHE-TIMEOUT*
              else
              ; Couldn't find servers? Try again soon
              60000)
              (fetch (NSMAILSTATE STATETIMER)
              of STATE])
      (if [AND NEWSERVERS (OR (NOT (EQ (LENGTH NEWSERVERS)
                                      (LENGTH OLDSERVERS))
                              (for SERVER in OLDSERVERS as PAIR in NEWSERVERS
                                  thereis (OR (NOT (EQUAL.CH.NAMES (CAR PAIR)
                                                                    (fetch MAILSERVERNAME of SERVER))))
                              (NOT (for I from 0 to 4
                                  bind (SERVERADDR _ (fetch MAILPORT of SERVER))

```

```

(PAIRADDR _ (CADR PAIR))
always (EQ (\GETBASE SERVERADDR I)
         (\GETBASE PAIRADDR I))
then ;; Yes, mailbox info is different. Fix it up. Note that we do nothing if no mail servers were found. This is to avoid
;; screwing up when we failed to talk to a clearinghouse (since otherwise we would find ourselves with no servers,
;; hence nobody to wake up periodically and find out where the servers have moved to). If only CH.RETRIEVE.ITEM
;; could give us an error return in that case...
(replace (LAFITEMODEDATA MAILSERVERS) of OLDDATA with (\NSMAIL.MAKE.MAILSERVERS
NEWSERVERS FULLNAME
(fetch (LAFITEMODEDATA CREDENTIALS
)
)
of OLDDATA)))
T])

```

(NS.NEXTMESSAGE

```

[LAMBDA (MAILBOX) ; Edited 26-Jun-90 18:18 by jds
(PROG ((ENVELOPES (fetch (NSMAILBOX NSMAILENVTAIL) of MAILBOX)))
(SELECTQ ENVELOPES
(NIL ; First time, read all envelopes
(COND
([OR (fetch (NSMAILBOX NSMAILENVELOPES) of MAILBOX)
(NULL (SETQ ENVELOPES (\NSMAIL.READ.ENVELOPES MAILBOX)
(RETURN)))
(replace (NSMAILBOX NSMAILENVELOPES) of MAILBOX with ENVELOPES)
(replace (NSMAILBOX NSMAILENVTAIL) of MAILBOX with ENVELOPES))
(T ; Finished
(RETURN))
NIL)
(RETURN (CAR ENVELOPES]))

```

(\NSMAIL.READ.ENVELOPES

```

[LAMBDA (MAILBOX) ; Edited 26-Jun-90 18:19 by jds
(LET [(ENVELOPES (INBASKET.CALL MAILBOX 'LIST (fetch (NSMAILBOX NSMAILSESSION) of MAILBOX)
(COURIER.CREATE (INBASKET . RANGE)
FIRST _ (fetch (NSMAILBOX NSMAILFIRSTINDEX) of MAILBOX)
LAST _ (fetch (NSMAILBOX NSMAILLASTINDEX) of MAILBOX))
(COURIER.CREATE (INBASKET . SELECTIONS)
TRANSPORT.ENVELOPE _ T INBASKET.ENVELOPE _ T MAIL.ATTRIBUTES _
(LIST (\NSMAIL.ATTRIBUTE.TYPE BodyType)))
' (INBASKET . MESSAGE.DESRIPTION]
for E in ENVELOPES collect (CONS (COURIER.FETCH (INBASKET . MESSAGE.DESRIPTION)
MESSAGE.INDEX of E)
(APPEND (COURIER.FETCH (INBASKET . MESSAGE.DESRIPTION)
CONTENT.ATTRIBUTES of E)
(COURIER.FETCH (INBASKET . MESSAGE.DESRIPTION)
TRANSPORT.ENVELOPE of E)
(COURIER.FETCH (INBASKET . MESSAGE.DESRIPTION)
INBASKET.ENVELOPE of E])

```

(INBASKET.CALL

```

[CL:LAMBDA (MAILBOX PROCEDURE &REST ARGS) ; Edited 26-Jun-90 18:19 by jds
(PROG ((STREAM (fetch (NSMAILBOX NSMAILSTREAM) of MAILBOX))
RESULT)
LP (if (AND (EQ [CAR (LISTP (SETQ RESULT (CL:APPLY (FUNCTION COURIER.CALL)
STREAM
'INBASKET PROCEDURE ARGS]
'ERROR)
(CASE (CAR (LAST ARGS))
(NOERROR NIL)
(RETURNERRORS ; We'll only handle stream lost--caller gets the rest
(EQ (CADR RESULT)
'STREAM.LOST))
(T ; Probably an error was already signaled
T)))
then (SETQ STREAM (\NSMAIL.SIGNAL.ERROR RESULT MAILBOX 'INBASKET PROCEDURE))
(GO LP)
else (RETURN RESULT])

```

(NS.RETRIEVEMESSAGE

```

[LAMBDA (MAILBOX MSGOUTFILE) ; Edited 26-Jun-90 18:19 by jds
(LET ((*RETRIEVAL-ERROR* NIL)
(ENVELOPE (pop (fetch (NSMAILBOX NSMAILENVTAIL) of MAILBOX)))
TYPE)
(if (OR NSMAIL.LEAVE.ATTACHMENTS (MEMB (SETQ TYPE (CADR (ASSOC 'BodyType ENVELOPE)))
\NSMAIL.GOOD.BODYTYPES))
then ; Retrieve ordinary text message, or retrieve the text part and
; leave attachment behind
(\NSMAIL.RETRIEVE MAILBOX ENVELOPE [FUNCTION (LAMBDA (MSGSTREAM)
;; MSGSTREAM is a bulk data stream containing content of msg.
;; as a 'serialized file'
(SETFILEINFO MSGSTREAM 'ENDOFSTREAMOP

```



```

(FUNCTION \NSMAIL.EOF.ON.RETRIEVE))
(NSMAIL.CHECK.SERIALIZED.VERSION MSGSTREAM)
(NSMAIL.READ.SERIALIZED.TREE MSGSTREAM
MSGOUTFILE (CDR ENVELOPE])

(GETFILEPTR MSGOUTFILE)
MSGOUTFILE)
(COND
(*RETRIEVAL-ERROR* (printout MSGOUTFILE T *RETRIEVAL-ERROR* T)))
else
; Not text or mail note, so retrieve the whole thing raw and make
; an "attachment"
(SETQ TYPE (\TYPE.FROM.FILETYPE TYPE))
(LET ((BUFFER (OPENSTREAM '{NODIRCORE} 'BOTH))
BODY ATTACHPOINT ATTRIBUTE.END)
[SETQ BODY (\NSMAIL.RETRIEVE MAILBOX ENVELOPE
(FUNCTION (LAMBDA (BULKSTREAM)
; Just eat it raw
(LET [(BODY (OPENSTREAM '{NODIRCORE} 'BOTH NIL
'(ENDOFSTREAMOP \NSMAIL.EOF.ON.RETRIEVE]
(COPYBYTES BULKSTREAM BODY)
BODY]
(SETFILEPTR BODY 0)
(NSMAIL.CHECK.SERIALIZED.VERSION BODY)
(NSMAIL.READ.SERIALIZED.TREE BODY BUFFER (CDR ENVELOPE)
T)
(SETQ ATTRIBUTE.END (GETFILEPTR BODY))
(SETQ BUFFER (OPENTEXTSTREAM BUFFER NIL NIL NIL (LIST 'FONT LAFITEDISPLAYFONT)))
(TEDIT.INSERT.OBJECT (\MAILOBJ.CREATE BODY TYPE ATTRIBUTE.END)
BUFFER
(if (SETQ ATTACHPOINT (TEDIT.FIND BUFFER "
Attachment: " 1))
then
; Insert object at end of this line
(+ ATTACHPOINT 14)
else
; Shouldn't happen
(+ (TEDIT.FIND BUFFER "
" 1)
2)))
(COPYBYTES (OPENSTREAM (COERCETEXTOBJ BUFFER 'FILE)
'INPUT)
MSGOUTFILE)
; Would like this to be (COERCETEXTOBJ BUFFER (QUOTE
; FILE) MSGOUTFILE) but Tedit has a bug
))
(COND
((NEQ (CADR ENVELOPE)
'NO)
; Read okay, tell close mailbox to delete it. NO set when there is
; an attachment to leave behind
(RPLACA (CDR ENVELOPE)
'DELETE]))

```

(\NSMAIL.RETRIEVE

```

[LAMBDA (MAILBOX ENVELOPE RETRIEVEFN START MSGOUTFILE)
; Edited 26-Jun-90 18:19 by jds
;; Perform an Inbasket.Retrieve on the specified message, using RETRIEVEFN to read the bulk data. If START is true, then the file pointer on
;; MSGOUTFILE is returned to START if we have to retry
(bind RESULT while (EQ [CAR (LISTP (SETQ RESULT (COURIER.CALL (fetch (NSMAILBOX NSMAILSTREAM) of MAILBOX)
'INBASKET
'RETRIEVE
(fetch (NSMAILBOX NSMAILSESSION) of MAILBOX)
(CAR ENVELOPE)
\NSMAIL.CTSTANDARD.MESSAGE RETRIEVEFN 'RETURNERRORS]
'ERROR)
do
; Maybe lost the stream?
(NSMAIL.SIGNAL.ERROR RESULT MAILBOX 'INBASKET 'RETRIEVE)
(AND START (SETFILEPTR MSGOUTFILE START))
finally (RETURN RESULT)])

```

(\NSMAIL.EOF.ON.RETRIEVE

```

[LAMBDA (STREAM)
(DECLARE (USEDFFREE *RETRIEVAL-ERROR*))
; Edited 9-Sep-88 12:29 by bvm
(SETQ *RETRIEVAL-ERROR* "***Warning: errors in message format**")
(COND
(LAFITEDEBUGFLG (HELP "EOF during retrieve")))
(LET (POS)
(COND
((SETQ POS (STKPOS (FUNCTION \NSMAIL.READ.SERIALIZED.TREE)))
(REFFROM POS NIL T))
(T 0))

```

(\NSMAIL.READ.SERIALIZED.TREE

```

[LAMBDA (MSGSTREAM MSGOUTFILE ENVELOPE ATTACHMENT)
; Edited 17-Jan-89 17:30 by bvm

```

;;; Read a message, which is in the format of a NS Filing Serialized File. This is the recursive part, SerializedTree. Format is --- Sequence of Attribute;
;;; Content; children = Sequence of SerializedTree


```

(COURIER.ABORT.BULKDATA))
(\NSMAIL.READ.SERIALIZED.CONTENT MSGSTREAM MSGOUTFILE)
(RPTQ (\WIN MSGSTREAM) ; Read children
(\NSMAIL.READ.SERIALIZED.TREE MSGSTREAM MSGOUTFILE])

```

(\NSMAIL.CHECK.SERIALIZED.VERSION

```

[LAMBDA (STREAM) ; Edited 5-May-89 14:47 by bvm
(LET [(V (COURIER.READ STREAM NIL 'LONGCARDINAL)
(SELECTC V
(\SERIALIZED.FILE.VERSIONS
T)
(HELP (CL:FORMAT NIL "Lafite does not understand serialized file version ~D.
RETURN to attempt retrieval anyway." V])

```

(\NSMAIL.READ.SERIALIZED.CONTENT

```

[LAMBDA (INSTREAM OUTSTREAM) ; Edited 17-Jan-89 17:14 by bvm

```

;; Interprets INSTREAM as SerializedTree.Content, i.e., as a Bulkdata.StreamOfUnspecified followed by the lastBytelsSignificant flag. Copies the raw data therein to OUTSTREAM

```

(bind LASTSEGMENT? BYTE BYTECOUNT do (SETQ LASTSEGMENT? (NEQ (\WIN INSTREAM)
0))
(COND
[(NEQ (SETQ BYTECOUNT (UNFOLD (\WIN INSTREAM)
BYTESPERWORD))
0)
(RPTQ (SUB1 BYTECOUNT)
(\BOUT OUTSTREAM (\BIN INSTREAM)))
(SETQ BYTE (\BIN INSTREAM))
; Final byte of this segment. Don't copy until we know whether
; it's significant
(COND
((OR (NULL LASTSEGMENT?)
(NEQ (\WIN INSTREAM)
0)) ; Not last segment, or the word after says the final byte was
; significant
(\BOUT OUTSTREAM BYTE)
(LASTSEGMENT? ; Null body. Throw out the lastBytelsSignificant flag
(\WIN INSTREAM)))
repeatuntil LASTSEGMENT?])

```

(\NSMAIL.DISCARD.SERIALIZED.CONTENT

```

[LAMBDA (INSTREAM) ; Edited 17-Jan-89 17:17 by bvm

```

;; Interprets INSTREAM as SerializedTree.Content, i.e., as a Bulkdata.StreamOfUnspecified followed by the lastBytelsSignificant flag and discards it all

```

(do (if (NEQ (PROG1 (\WIN INSTREAM)
(RPTQ (UNFOLD (\WIN INSTREAM)
BYTESPERWORD)
(\BIN INSTREAM)))
0)
then ; Finished. Read the lastBytelsSignificant flag
(\WIN INSTREAM)
(RETURN])

```

(\NSMAIL.READ.STRING.AS.STREAM

```

[LAMBDA (INSTREAM OUTSTREAM) (* bvm%: "30-Jul-84 16:13")

```

;; Considers INSTREAM to be positioned at a sequence of unspecified, and reads it as if its datatype were string, and copies said bytes to OUTSTREAM

```

(PROG (LENGTH)
(\WIN INSTREAM) ; Skip sequence count
(COPYBYTES INSTREAM OUTSTREAM (SETQ LENGTH (\WIN INSTREAM)))
(COND
((ODDP LENGTH)
(\BIN INSTREAM])

```

(\NSMAIL.PRINT.HEADERFIELDS

```

[LAMBDA (MSGOUTFILE HEADERFIELDS ENVELOPE SENDER NOTEBODY ATTACHMENT DISCARDED) ; Edited 4-Aug-89 18:34 by bvm

```

;; Compose message header from HEADERFIELDS and ENVELOPE, printing to MSGOUTFILE. SENDER is the "Sender" field of the message, if we encountered one, or sole element of the "From" field. NOTEBODY if non-NIL is a stream containing the text of a Note attribute. if ATTACHMENT is true, we add a line "Attachment:" to the message where caller will later insert the attachment object. DISCARDED is list of fields we didn't recognize.

```

(LET
(TYPE BADNAMES REASON TMP VALUE ID)
(SETQ HEADERFIELDS (REVERSE HEADERFIELDS))
(COND
(ENVELOPE [if (SETQ VALUE (ASSOC 'TransportProblem ENVELOPE))
then ; Return of undeliverable mail
(SETQ HEADERFIELDS (DREMOVE VALUE HEADERFIELDS))

```


X)
(NULL (FMEMB (CAR Y)
NSMAIL.HEADER.ORDER])

```
when (SETQ VALUE (CDR PAIR))
do
(printout MSGOUTFILE (SETQ TYPE (CAR PAIR))
": ")
(CASE TYPE
(Date (printout MSGOUTFILE (GDATE VALUE (DATEFORMAT NO.SECONDS TIME.ZONE SPACES))))
((From To cc Reply-to)
(\NSMAIL.PRINT.NAMES
VALUE MSGOUTFILE
(SELECTQ TYPE
(From
(if [AND SENDER (NOT (for NAME in VALUE
always (OR (EQ NAME SENDER)
(AND (STRING-EQUAL (fetch NSDOMAIN of NAME)
(fetch NSDOMAIN of SENDER))
(STRING-EQUAL (fetch NSORGANIZATION of NAME)
(fetch NSORGANIZATION of SENDER])
then
; Ugh, From and Sender are different domains. To reduce
; confusion, force everything to be fully qualified
(SETQ SENDER NIL))
NIL)
(Reply-to
; always full-qualified
SENDER)))
((Sender Originator) (printout MSGOUTFILE (NSNAME.TO.STRING VALUE T)))
(Attachment
(printout MSGOUTFILE "%Type " .I1 VALUE " ID " .P2 ID "%")
(RPLACA ENVELOPE 'NO))
(T (while (AND (> (NCHARS VALUE)
0)
(EQ (NTHCHARCODE VALUE -1)
(CHARCODE CR)))
do
; Trailing cr's, e.g., in the Subject line, will cause the header not
; to parse
(SETQ VALUE (SUBSTRING VALUE 1 -2)))
(if (STRPOS "
" VALUE)
then
; Internal CR? I suppose we could print it and make sure there is
; whitespace at the start of the next line, but why bother?
(SETQ VALUE (CL:SUBSTITUTE #\ \ #\Newline VALUE)))
(PRIN1 VALUE MSGOUTFILE))
(TERPRI MSGOUTFILE))
(if DISCARDED
then (printout MSGOUTFILE "Discarded-Fields: ")
(LA.PRINT.COMMA.LIST (REVERSE DISCARDED)
MSGOUTFILE)
(TERPRI MSGOUTFILE))
(COND
(ATTACHMENT
; Reserve a line where the attachment object will be placed.
(PRINTOUT MSGOUTFILE T "Attachment: " T)))
(COND
(NOTEBODY (TERPRI MSGOUTFILE)
(COPYBYTES NOTEBODY MSGOUTFILE 0 -1)
(TERPRI MSGOUTFILE))
```

(\NSMAIL.PRINT.NAMES

; Edited 5-Jan-90 18:30 by bvm

```
[LAMBDA (NSNAMES OUTSTREAM DEFAULTNAME)
(for NAME in NSNAMES bind (FIRSTTIME _ T)
ORGDIFFERS
do (COND
(FIRSTTIME (SETQ FIRSTTIME NIL))
(T (PRIN3 ", " OUTSTREAM)))
(PRIN3 (fetch NSOBJECT of NAME)
OUTSTREAM)
(LET ((ORG (fetch NSORGANIZATION of NAME))
(DOM (fetch NSDOMAIN of NAME)))
(if [OR [SETQ ORGDIFFERS (NOT (AND DEFAULTNAME (OR (STRING-EQUAL ORG (fetch NSORGANIZATION
of DEFAULTNAME))
(EQ (NCHARS ORG)
0])
(NOT (OR (STRING-EQUAL DOM (fetch NSDOMAIN of DEFAULTNAME))
(EQ (NCHARS DOM)
0])
then
;; Have to print the domain. The null string tests are because there exists buggy software that doesn't fill in the domain
;; and org--we want them to default correctly eventually.
(PRIN3 ":" OUTSTREAM)
(PRIN3 DOM OUTSTREAM)
(if ORGDIFFERS
then
; Have to print the org, too
(PRIN3 ":" OUTSTREAM)
(PRIN3 ORG OUTSTREAM])
```

)

:: Error handling

(DEFINEQ

(\NSMAIL.COURIER.OPEN

[LAMBDA (ADDRESS) ; Edited 9-Sep-88 12:06 by bvm
(COURIER.OPEN ADDRESS NIL T 'NSMAIL NIL (CONSTANT (LIST 'ERRORHANDLER (FUNCTION \NSMAIL.ERRORHANDLER))

(\NSMAIL.ERRORHANDLER

[LAMBDA (STREAM ERRCODE) ; Edited 9-Sep-88 12:35 by bvm

:: Called when SPP error occurs on NS mail courier connection STREAM. Fakes an error return from the courier.call.

(LET (POS)
(if [AND (EQ ERRCODE 'STREAM.LOST)
(SETQ POS (STKPOS (FUNCTION COURIER.CALL)
(BLOCK 500)
then (RETFROM POS ' (ERROR STREAM.LOST)
T)
else (\SPP.DEFAULT.ERRORHANDLER STREAM ERRCODE)])

(\NSMAIL.SIGNAL.ERROR

[LAMBDA (ERROR MAILBOX PROGRAM PROCEDURE) ; Edited 26-Jun-90 18:19 by jds

:: Called when we get an error on an NS mail courier call. If stream lost, then tries to reestablish the connection, returning a new stream on
:: success.

(if (EQ (CADR ERROR)
'STREAM.LOST)
then (PRINTOUT PROMPTWINDOW T "Lost NS mail connection, trying to reestablish...")
(LET [(STREAM (\NSMAIL.COURIER.OPEN (create NSADDRESS using (SPP.DESTADDRESS (fetch (NSMAILBOX
NSMAILSTREAM
)
of MAILBOX))
NSSOCKET _ 0]
(if STREAM
then (PRINTOUT PROMPTWINDOW "done.")
(replace (NSMAILBOX NSMAILSTREAM) of MAILBOX with STREAM)
else (PRINTOUT PROMPTWINDOW "failed.")
(ERROR "NS mail connection lost, can't reestablish"))
else (COURIER.SIGNAL.ERROR PROGRAM PROCEDURE ERROR)])

)

:: Close/flush protocol

(DEFINEQ

(\NS.CLOSEMAILBOX

[LAMBDA (MAILBOX FLUSH?) ; Edited 26-Jun-90 18:19 by jds

[COND (FLUSH?) ; Mark everything either deleted or seen

(for E in (fetch (NSMAILBOX NSMAILENVELOPES) of MAILBOX) bind START STATUS
do [COND
((NEQ (CADR E)
STATUS)
(COND
(START (\NSMAIL.CHANGE.STATUS MAILBOX START (SUB1 (CAR E))
STATUS)))
(SETQ START (CAR E))
(SETQ STATUS (CADR E))
finally (COND
(START (\NSMAIL.CHANGE.STATUS MAILBOX START (fetch (NSMAILBOX NSMAILLASTINDEX)
of MAILBOX)
STATUS]
(\NSMAIL.LOGOFF (fetch (NSMAILBOX NSMAILSTATE) of MAILBOX)
(fetch (NSMAILBOX NSMAILSTREAM) of MAILBOX)])

(\NSMAIL.LOGOFF

[LAMBDA (STATE STREAM) ; Edited 26-Jun-90 18:22 by jds

:: Executes the Inbasket.Logoff procedure and clears appropriate state. Returns true if LOGOFF call succeeded.

(LET [(RESULT (COURIER.CALL STREAM 'INBASKET 'LOGOFF (fetch (NSMAILSTATE STATESESSION) of STATE)
'RETURNERRORS]
(PROG1 (AND (LISTP RESULT)
(NEQ (CAR RESULT)
'ERROR))
(replace (NSMAILSTATE STATESESSION) of STATE with NIL)
:: Once session is closed, can't say anything about first new message if there are any messages left, because someone in the
:: meantime could delete them from another session
(replace (NSMAILSTATE STATEFIRSTNEW) of STATE with NIL)
(replace (NSMAILSTATE STATEOLDLAST) of STATE with NIL)

(CLOSEF STREAM))]]

(\NSMAIL.CHANGE.STATUS

[LAMBDA (MAILBOX START END STATUS)

; Edited 26-Jun-90 18:19 by jds

:: Change status of messages START thru END to be STATUS, which is either DELETE or KEEP. Returns number of messages kept

```
(PROG ((SESSION (fetch (NSMAILBOX NSMAILSESSION) of MAILBOX))
(STREAM (fetch (NSMAILBOX NSMAILSTREAM) of MAILBOX))
(RANGE (COURIER.CREATE (INBASKET . RANGE)
FIRST _ START LAST _ END)))
(RETURN (COND
((EQ STATUS 'DELETE)
(COURIER.CALL STREAM 'INBASKET 'DELETE SESSION RANGE)
0)
(T (COURIER.CALL STREAM 'INBASKET 'CHANGE.STATUS SESSION RANGE 'KNOWN)
(ADD1 (IDIFFERENCE END START))
))
))
```

(RPAQ? NSMAILDEBUGFLG)

(RPAQ? NSMAIL.LEAVE.ATTACHMENTS)

(RPAQ? NSMAIL.HEADER.ORDER ' (Date Sender From Subject In-Reply-to To cc Message-ID Reply-to))

(ADDTOVAR \NSMAIL.GOOD.BODYTYPES 2 4)

:: Handling attachments as a special kind of image object

(DEFINEQ

(\MAILOBJ.CREATE

[LAMBDA (DATA TYPE ATTR.LENGTH NAME MORE.INFO START)

; Edited 14-Feb-90 16:59 by bvm

:: Create a mail object encapsulating data (a core file in serialized file format). TYPE is the type of the serialized data.

```
(OR START (SETQ START 0))
(LET* ([TITLE (SELECTQ TYPE
(REFERENCE
(if (NOT MORE.INFO)
then
(LET* [(INFO (CADR (\MAILOBJ.PARSE.ATTRIBUTES DATA (LIST
MAILOBJ.REFERENCE.FIELD
)
START)))
(TYPE (\TYPE.FROM.FILETYPE (CADR (ASSOC 'TYPE INFO)
[SETQ NAME (\MAILOBJ.NS.TO.LISP.NAME
(CADR (ASSOC 'HOST INFO))
(CADR (ASSOC 'DIRECTORY INFO))
(CADR (ASSOC 'NAME INFO))
(AND (NEQ (CADR (ASSOC 'FLAGS INFO))
\MAILOBJ.REFERENCE.LAST.FILED)
(CADR (ASSOC 'VERSION INFO))
(EQ TYPE 'DIRECTORY]
(SETQ MORE.INFO `(FILE.ID ,(CADR (ASSOC 'FILE.ID INFO))
TYPE
,TYPE]
(CL:FORMAT NIL "Reference to ~A ~A" (\MAILOBJ.TYPE.NAME (LISTGET MORE.INFO
'TYPE))
NAME))
(if NAME
then (CONCAT NAME " (" (\MAILOBJ.TYPE.NAME TYPE T)
")")
else (\MAILOBJ.TYPE.NAME TYPE]
(TITLELEN (NCHARS TITLE))
[FONT (AND (> TITLELEN 20)
(LET* [(FONT DEFAULTICONFONT)
(SIZE (FONTPROP FONT 'SIZE)
; Use a smaller font if available
(if (> TITLELEN 30)
then
; This is really getting out of hand..
(SETQ TITLE (CONCAT (SUBSTRING TITLE 1 25)
"..."))
(AND (> SIZE 8)
(CAR (NLSETQ (FONTCOPY FONT 'SIZE (- SIZE 2]
(IMAGE (WINDOWPROP (TITLEDICONW NIL TITLE FONT '(0 . 0)
T NIL 'FILE)
'ICONIMAGE]
; Crude way of getting a bitmap with some text printed on it
; nicely
(IMAGEOBJCREATE [create MAILOBJ
MAILOBJ.IMAGE _ IMAGE
MAILOBJ.BOX _ (create IMAGEBOX
XSIZE _ (BITMAPWIDTH IMAGE)
YSIZE _ (BITMAPHEIGHT IMAGE)
YDESC _ (LRSH (BITMAPHEIGHT IMAGE)
1)
XKERN _ 0)
```

```

MAILOBJ.TYPE _ TYPE
MAILOBJ.DATA _ DATA
MAILOBJ.ATTR.LENGTH _ ATTR.LENGTH
MAILOBJ.START _ START
MAILOBJ.NAME _ NAME
MAILOBJ.INFO _ MORE.INFO
MAILOBJ.EXPANDABLE _ (PROGN ; True if object has children
(CADR (MAILOBJ.PARSE.ATTRIBUTES
DATA
(CONSTANT (LIST (ASSOC 'IS.DIRECTORY
\NSFILING.ATTRIBUTES)
))
START]

```

\MAILOBJ.IMAGEFNS])

(MAILOBJ.TYPE.NAME

```

[LAMBDA (TYPE SHORT) ; Edited 29-Sep-87 14:21 by bvm:
;; Translate filing TYPE into a descriptive string, e.g., "Interpress Document". If SHORT is true, leave out "Document". If TYPE is numeric, it is
;; rendered as "Type nnn Document".
(if (EQ TYPE 'DIRECTORY)
then ; Viewpoint calls these "folders"
"Viewpoint Folder"
else (CL:FORMAT NIL "~:[~:(~A~)~;Type ~D~]~@[ Document~]" (FIXP TYPE)
TYPE
(NOT SHORT])

```

(MAILOBJ.NS.TO.LISP.NAME

```

[LAMBDA (HOST DIRECTORY NAME VERSION DIRECTORYFLG) ; Edited 29-Sep-87 17:54 by bvm:
;; Turn these pieces parsed out of a reference icon into a Lisp-style file name. Mainly this means turning the slashes into angles. This code is
;; stolen from \NSFILING.FULLNAME, which is what we would use if it didn't require a filing session arg.
(LET ((PATHNAME (if DIRECTORYFLG
then (CONCAT DIRECTORY "/" NAME (if (AND VERSION (NEQ VERSION 1))
then (CONCAT "!" VERSION)
else ""))
else DIRECTORY))
FILENAME DIRLST FULLNAME FUNNYCHAR DOTSEEN QUOTEDDIRS)
(for I from 1 bind CH (START _ 1) while (SETQ CH (NTHCHARCODE PATHNAME I))
do (SELCHARQ CH
(%' ; quote mark, skip it and next char
(add I 1))
(/ ; Directory marker
(push DIRLST (SUBSTRING PATHNAME START (SUB1 I)))
(SETQ START (ADD1 I)))
((; %: < > } %]) ; Funny characters that filing doesn't care about but we do --
; need to quote these
(SETQ FUNNYCHAR T))
NIL)
finally (push DIRLST (SUBSTRING PATHNAME START)))
;; DIRLST is in reverse order now.
(for DIR in DIRLST do (push QUOTEDDIRS (COND
(FUNNYCHAR (\NSFILING.ADDQUOTES DIR T))
(T DIR))
'>))
(CONCATLIST (NCONC (LIST '{ HOST "<">
QUOTEDDIRS
(AND (NOT DIRECTORYFLG)
(CONS (\NSFILING.ADDQUOTES NAME)
(AND VERSION (LIST (if (STRPOS "." NAME)
then ".";
else ".;")
VERSION]))

```

(MAILOBJ.DISPLAY

```

[LAMBDA (OBJ STREAM) ; Edited 26-Jun-90 18:17 by jds
(LET [(IMAGE (fetch (MAILOBJ MAILOBJ.IMAGE) of (fetch OBJECTDATUM of OBJ)] ; Display the image, centered on the baseline
(BITBLT IMAGE NIL NIL STREAM (DSPXPOSITION NIL STREAM)
(- (DSPYPOSITION NIL STREAM)
(LRSH (BITMAPHEIGHT IMAGE)
1])

```

(MAILOBJ.GET

```

[LAMBDA (STREAM TEXTSTREAM) ; Edited 14-Feb-90 16:50 by bvm
(DESTRUCTURING-BIND (LEN TYPE ATTR.LEN NAME . INFO)
(READ STREAM FILERDTBL)
(LET (DATASTREAM START)
(if (EQ (fetch DEVICENAME of (fetch (STREAM DEVICE) of STREAM))
'NODIRCORE)
then ; No need to copy the data, just copy the cover
(SETQ DATASTREAM (NCREATE 'STREAM STREAM))

```



```

      (SETQ START (GETFILEPTR STREAM))
      (LET ((EOF (+ START LEN))) ; Fix the eof so we don't have to carry around the length
        (replace (STREAM EPAGE) of DATASTREAM with (FOLDLO EOF BYTESPERPAGE))
        (replace (STREAM EOFSET) of DATASTREAM with (IMOD EOF BYTESPERPAGE)))
      else (SETQ DATASTREAM (OPENSTREAM '{NODIRCORE}' 'BOTH))
        (COPYBYTES STREAM DATASTREAM LEN)
        (SETQ START 0))
(\MAILOBJ.CREATE DATASTREAM TYPE ATTR.LEN NAME INFO START])

```

(\MAILOBJ.IMAGEBOX

```

[LAMBDA (OBJ) ; Edited 26-Jun-90 18:17 by jds
  (fetch (MAILOBJ MAILOBJ.BOX) of (fetch OBJECTDATUM of OBJ))

```

(\MAILOBJ.PUT

```

[LAMBDA (OBJ STREAM) ; Edited 26-Jun-90 18:17 by jds
  (LET* ((MAILOBJ (fetch OBJECTDATUM of OBJ))
        (COREFILE (fetch (MAILOBJ MAILOBJ.DATA) of MAILOBJ))
        (END (GETEOFPTR COREFILE))
        (START (fetch (MAILOBJ MAILOBJ.START) of MAILOBJ)))
    (LET ((*PRINT-BASE* 10)
          (*READTABLE FILERDTBL)
          (NAME (fetch (MAILOBJ MAILOBJ.NAME) of MAILOBJ))
          (INFO (fetch (MAILOBJ MAILOBJ.INFO) of MAILOBJ))) ; Make sure we can read it back.
      (PRIN4 (LIST* (- END START)
                   (fetch (MAILOBJ MAILOBJ.TYPE) of MAILOBJ)
                   (fetch (MAILOBJ MAILOBJ.ATTR.LENGTH) of MAILOBJ)
                   (AND (OR NAME INFO)
                        (CONS NAME INFO)))
            STREAM))
    (COPYBYTES COREFILE STREAM START END])

```

(\MAILOBJ.INIT

```

[LAMBDA NIL ; Edited 29-Jun-87 16:36 by bvm:
  (SETQ \MAILOBJ.IMAGEFNS (IMAGEFNSCREATE (FUNCTION \MAILOBJ.DISPLAY)
                                           (FUNCTION \MAILOBJ.IMAGEBOX)
                                           (FUNCTION \MAILOBJ.PUT)
                                           (FUNCTION \MAILOBJ.GET)
                                           (FUNCTION CL:IDENTITY)
                                           (FUNCTION \MAILOBJ.BUTTONEVENTFN))

```

(DEFINEQ

(\MAILOBJ.BUTTONEVENTFN

```

[LAMBDA (OBJ WINDOWSTREAM SELECTION RELX RELY WINDOW TEXTSTREAM BUTTON) ; Edited 26-Jun-90 18:17 by jds
  (if (.COPYKEYDOWNP.)
      then ; There's more to copy selection than this
        [AND NIL (LET [(NAME (fetch (MAILOBJ MAILOBJ.NAME) of (IMAGEOBJPROP OBJ 'OBJECTDATUM))
                          (AND NAME (BKSYSBUF NAME))
                        (PRINTOUT PROMPTWINDOW T "Attachment is busy")
                      ; Busy
                    else (LET* [(MAILOBJ (IMAGEOBJPROP OBJ 'OBJECTDATUM))
                              (TYPE (fetch (MAILOBJ MAILOBJ.TYPE) of MAILOBJ))
                              (REAL.TYPE (if (EQ TYPE 'REFERENCE)
                                             then (LISTGET (fetch (MAILOBJ MAILOBJ.INFO) of MAILOBJ)
                                                           'TYPE)
                                             else TYPE))
                            (CMD (MENU (create MENU
                                              ITEMS _
                                              `(("View as text" '\MAILOBJ.VIEW "View the attachment as raw text, using
                                              Tedit")
                                              (, (if (EQ TYPE 'REFERENCE)
                                                    then ; Note that we are storing the reference itself, not the referenced
                                                    ; file
                                                    "Store reference"
                                                    else "Put to file")
                                              '\MAILOBJ.PUT.FILE "Store the attachment in a file. This operation loses
                                              information unless the file is on an NS File Server.")
                                              ,@[AND (EQ REAL.TYPE 'INTERPRESS)
                                                  ' ("Send to Printer" '\MAILOBJ.HARDCOPY "Send the document to the
                                                  printer of your choice."]
                                              ,@[AND (fetch (MAILOBJ MAILOBJ.EXPANDABLE) of MAILOBJ)
                                                  ' ("Expand folder" '\MAILOBJ.EXPAND "Extract the first-level
                                                  subparts of the folder"]
                                              ,@(SELECTQ TYPE
                                                  (REFERENCE [AND (GETD 'FILEBROWSER)
                                                                (EQ (NTHCHARCODE (fetch (MAILOBJ MAILOBJ.NAME)
                                                                 of MAILOBJ)
                                                                -1)
                                                                (CHARCODE >))
                                                  `(("FileBrowse" '\MAILOBJ.FB "Invoke the File

```

Browser on the referenced object"])

```

NIL))
CENTERFLG _ T]
(if (NULL CMD)
  then
    T ; Nothing selected; allow TEdit to select
  else
    ; Do the command in its own process so that the window can
    ; return to its more natural state (instead of severely clipped)
    (ADD.PROCESS (LIST (FUNCTION \MAILOBJ.DO.COMMAND)
      (KWOTE CMD)
      (KWOTE OBJ)
      (KWOTE WINDOW)
      (KWOTE TEXTSTREAM))
      'NAME
      'MAILOBJ
      'RESTARTABLE
      'HARDRESET
      'BEFOREEXIT
      'DON'T) ; Return DON'T so that the window doesn't pop on top to select
    'DON'T])

```

(\MAILOBJ.DO.COMMAND

```

[LAMBDA (CMD OBJ WINDOW TEXTSTREAM) ; Edited 3-Jul-87 17:51 by bvm:
  (RESETLST
    (RETSAVE (IMAGEOBJPROP OBJ 'BUSY T)
      (LIST 'IMAGEOBJPROP OBJ 'BUSY NIL))
    (CL:FUNCALL CMD OBJ WINDOW TEXTSTREAM)))

```

(\MAILOBJ.HARDCOPY

```

[LAMBDA (OBJ WINDOW) ; Edited 26-Jun-90 18:17 by jds

```

;; Hardcopy the attachment in MAILOBJ. WINDOW is the window in which we are viewing it (not currently used).

```

(LET* ((*UPPER-CASE-FILE-NAMES* NIL)
  (PRINTER (GetPrinterName))
  (MAILOBJ (IMAGEOBJPROP OBJ 'OBJECTDATUM))
  (REFP (EQ (fetch (MAILOBJ MAILOBJ.TYPE) of MAILOBJ)
    'REFERENCE)))
  ATTRIBUTES PRINTRESULTS NAME DATA START)
  (if (NULL PRINTER)
    then ; abort
      NIL
    elseif (NOT (STRPOS ":" PRINTER))
      then ; not ns
        (PRINTOUT PROMPTWINDOW T PRINTER " is not an Interpress printer")
    else (SETQ PRINTER (GETNSPRINTER PRINTER))
      (if REFP
        then (NSPRINT PRINTER (SETQ NAME (fetch (MAILOBJ MAILOBJ.NAME) of MAILOBJ)))
        else ; Have to do this by hand, since we don't have a nice standalone
          ; stream
          [SETQ ATTRIBUTES (\MAILOBJ.PARSE.ATTRIBUTES
            (SETQ DATA (fetch (MAILOBJ MAILOBJ.DATA) of MAILOBJ))
            [CONSTANT `([DOCUMENT.NAME ,@(CDR (ASSOC 'NAME \NSFILING.ATTRIBUTES)
              (DOCUMENT.CREATION.DATE ,@(CDR (ASSOC 'CREATED.ON
                \NSFILING.ATTRIBUTES)
                (SETQ START (fetch (MAILOBJ MAILOBJ.START) of MAILOBJ))
                ; Parse out the name and creation date, and use them for the
                ; document name/date
            [if (SETQ NAME (LISTGET ATTRIBUTES 'DOCUMENT.NAME))
              then ; Fix up any wayward subject
                (LISTPUT ATTRIBUTES 'DOCUMENT.NAME (SETQ NAME (\MAILOBJ.MUNGE.NAME NAME))
            [SETQ PRINTRESULTS (\NSPRINT.INTERNAL PRINTER ATTRIBUTES
              (FUNCTION (LAMBDA (DATASTREAM)
                (\MAILOBJ.COPY.BODY DATA DATASTREAM
                  (+ START (fetch (MAILOBJ MAILOBJ.ATTR.LENGTH)
                    of MAILOBJ)))
                NIL]
            (if (AND PRINTRESULTS NSPRINT.WATCHERFLG)
              then ; Set up a 'watchdog' process to keep the guy informed of the
                ; print job's status.
                (\NSPRINT.WATCH.JOB PRINTRESULTS PRINTER NAME)))
            (PRINTOUT PROMPTWINDOW T NAME " sent to " (fetch NSOBJECT of (CAR PRINTER]))

```

(\MAILOBJ.FB

```

[LAMBDA (OBJ WINDOW) ; Edited 26-Jun-90 18:17 by jds

```

;; Invoke the File Browser on the referenced object

```

(FILEBROWSER (fetch (MAILOBJ MAILOBJ.NAME) of (IMAGEOBJPROP OBJ 'OBJECTDATUM]))

```

(\MAILOBJ.PUT.FILE

```

[LAMBDA (OBJ WINDOW) ; Edited 26-Jun-90 18:17 by jds

```

;; Store the attachment of MAILOBJ as file of user's choosing. Prompt for file name. If it's on an NS directory, we can deserialize and thus
;; preserve the whole thing.

```
(LET*
  ((MAILOBJ (IMAGEOBJPROP OBJ 'OBJECTDATUM))
   (DATA (fetch (MAILOBJ MAILOBJ.DATA) of MAILOBJ))
   (START (fetch (MAILOBJ MAILOBJ.START) of MAILOBJ))
   (PW (CREATEW (create REGION
                  LEFT _ LASTMOUSEX
                  BOTTOM _ LASTMOUSEY
                  WIDTH _ (WINDOWPROP WINDOW 'WIDTH)
                  HEIGHT _ (HEIGHTIFWINDOW (TIMES 4 (FONTPROP DEFAULTFONT 'HEIGHT))
                                       NIL 8))
               NIL 8))
   FILE DEVICE CONDITION)
  (if [NULL (SETQ FILE (TTYINPROMPTFORWARD "Put attachment to file: " NIL NIL PW NIL 'TTY (CHARCODE (CR)
    then (PRINTOUT PW "...aborted")
  elseif (NULL (SETQ DEVICE (\GETDEVICEFROMNAME (SETQ FILE (\ADD.CONNECTED.DIR FILE))
    T)))
    then (PRINTOUT PW T "No such server/device")
  else (ALLOW.BUTTON.EVENTS)
    (PRINTOUT PW "... ")
    (if [CL:MULTIPLE-VALUE-SETQ (FILE CONDITION)
      (IGNORE-ERRORS (if (EQ (fetch OPENFILE of DEVICE)
        (FUNCTION \NSFILING.OPENFILE))
          then ;NS device. Really need better test than this.
            (SETFILEPTR DATA START)
            (LET ((*UPPER-CASE-FILE-NAMES* NIL))
              (DECLARE (CL:SPECIAL *UPPER-CASE-FILE-NAMES*))
                ;Get name pretty
              (\NSFILING.DESERIALIZE FILE DATA DEVICE))
            else [SETQ FILE (OPENSTREAM FILE 'OUTPUT 'NEW
              '(TYPE , (fetch (MAILOBJ MAILOBJ.TYPE) of MAILOBJ))
              (SEQUENTIAL T)
              (PRINTOUT PW "(some attributes will be lost) ")
              (\MAILOBJ.COPY.BODY DATA FILE (+ START (fetch (MAILOBJ
                MAILOBJ.ATTR.LENGTH
                )
              of MAILOBJ))
              PW)
            (CLOSEF FILE)
          then (PRINTOUT PW T FILE " written.")
        else (PRINTOUT PW "failed: " CONDITION)])
```

(\MAILOBJ.VIEW

[LAMBDA (OBJ WINDOW)

; Edited 26-Jun-90 18:17 by jds

:: View the text of the attachment. This is often enough to tell you whether you want to bother doing something more exciting with it.

```
(RESETLST
  [LET*
    ((MAILOBJ (IMAGEOBJPROP OBJ 'OBJECTDATUM))
     (TYPE (fetch (MAILOBJ MAILOBJ.TYPE) of MAILOBJ))
     (REFP (EQ TYPE 'REFERENCE))
     (WREG (WINDOWREGION (OR (CAR (WINDOWPROP WINDOW 'EXTRAWINDOWS))
                          WINDOW)))
     PROPS W SUBJECT START DATA DATASTART)
    [if REFP
      then (SETQ SUBJECT (fetch (MAILOBJ MAILOBJ.NAME) of MAILOBJ))
            (SETQ TYPE (LISTGET (fetch (MAILOBJ MAILOBJ.INFO) of MAILOBJ)
                              'TYPE))
            (SETQ START NIL)
      else (SETQ DATA (fetch (MAILOBJ MAILOBJ.DATA) of MAILOBJ))
            [SETQ SUBJECT (CADR (\MAILOBJ.PARSE.ATTRIBUTES DATA (CONSTANT (LIST (ASSOC 'NAME
              \NSFILING.ATTRIBUTES))
              )
            (SETQ DATASTART (fetch (MAILOBJ MAILOBJ.START) of MAILOBJ))
            (SETQ START (+ DATASTART (fetch (MAILOBJ MAILOBJ.ATTR.LENGTH) of MAILOBJ))
            [SETQ W (CREATEW (create REGION using WREG LEFT _ (+ (fetch (REGION LEFT) of WREG)
              (if (> (+ (fetch (REGION LEFT) of WREG)
                (fetch (REGION WIDTH) of WREG)
                MAILOBJ.WINDOWOFFSET)
                SCREENWIDTH)
              then (- MAILOBJ.WINDOWOFFSET)
            else MAILOBJ.WINDOWOFFSET))
            BOTTOM _ (- (fetch (REGION BOTTOM) of WREG)
              (if (< (- (fetch (REGION BOTTOM) of WREG)
                MAILOBJ.WINDOWOFFSET)
                0)
              then (- MAILOBJ.WINDOWOFFSET)
            else MAILOBJ.WINDOWOFFSET)))
            (CONCAT "Attachment: " (\MAILOBJ.MUNGE.NAME SUBJECT)
              ; Make window slightly overlapping display window
            (WINDOWADDPROP WINDOW 'EXTRAWINDOWS W T)
            [if (NEQ TYPE 'TEDIT)
              then ; TEdit's not so good on binary files, so just pull out the text.
                (LET [(COMPACTDATA (OPENSTREAM '{NODIRCORE} 'BOTH)
                  [if REFP
                    then [RESETSAVE NIL (LIST 'CLOSEF (SETQ DATA (OPENSTREAM SUBJECT 'INPUT NIL
                      ' ((SEQUENTIAL T)
```


(\MAILOBJ.COPY.CHILD

```
[LAMBDA (INSTREAM OUTSTREAM) ; Edited 6-Jul-87 14:41 by bvm:
;; This is the counterpart to \nsmail.read.serialized.tree, except that it copies the data as it parses it, rather than interpreting it. Returns file pointer
;; of the start of the main child's data section.
;; We are parsing here the recursive part of Filing.SerializedFile: SerializedTree, which consists of: Sequence of Attribute; Content; children =
;; Sequence of SerializedTree
(LET (ATTRLENGTH SUBSTART NCHILDREN LASTSEGMENT?)
  (\WOUT OUTSTREAM (SETQ ATTRLENGTH (\WIN INSTREAM))) ; number of attributes
  (to ATTRLENGTH do (RPTQ 4 (\BOUT OUTSTREAM (\BIN INSTREAM)))
    (\MAILOBJ.COPY.SEQUENCE INSTREAM OUTSTREAM) ; Copy attribute type (longcardinal)
    ; Copy attribute value (sequence unspecified)
  )
  (SETQ SUBSTART (GETFILEPTR OUTSTREAM))
  ;; Now copy the body, which is StreamOfUnspecified followed by lastBytelsSignificant boolean
  (do (\WOUT OUTSTREAM (SETQ LASTSEGMENT? (\WIN INSTREAM)))
    (\MAILOBJ.COPY.SEQUENCE INSTREAM OUTSTREAM) ; 1 => this is last segment
    finally (\WOUT OUTSTREAM (\WIN INSTREAM)) ; Copy the sequence repeatuntil (NEQ LASTSEGMENT? 0)
    ; Copy lastBytelsSignificant boolean
  )
  (\WOUT OUTSTREAM (SETQ NCHILDREN (\WIN INSTREAM)))
  (to NCHILDREN do (\MAILOBJ.COPY.CHILD INSTREAM OUTSTREAM))
  SUBSTART])
```

(\MAILOBJ.COPY.SEQUENCE

```
[LAMBDA (INSTREAM OUTSTREAM) ; Edited 6-Jul-87 14:37 by bvm:
;; Copy a Sequence of Unspecified from in to out.
(LET ((SEQLLENGTH (\WIN INSTREAM)))
  (\WOUT OUTSTREAM SEQLLENGTH) ; Representation is sequence length (word) followed by that
  ; many words
  (RPTQ (UNFOLD SEQLLENGTH BYTESPERWORD)
    (\BOUT OUTSTREAM (\BIN INSTREAM]))
```

(\MAILOBJ.EXTRACT.TEXT

```
[LAMBDA (DATA OUTSTREAM LEN) ; Edited 15-Aug-89 16:38 by bvm
;; Copy LEN bytes from the stream DATA to OUTSTREAM, where all the runs of non-printing characters are replaced by some small number of
;; ugly characters that won't upset teddit.
(to LEN bind CH HELDCH (SKIPPING _ -1) do (if (OR (>= (SETQ CH (\BIN DATA))
127)
(AND (< CH (CHARCODE SPACE))
(SELCHARQ CH
((TAB CR)
NIL)
( ; VP eol
(SETQ CH (CHARCODE CR))
NIL)
T)))
then ; Junk
(SETQ HELDCH NIL)
; I don't care if the previous byte was accidentally ascii
(if (EVENP (add SKIPPING 1)
16)
then (BOUT OUTSTREAM MAILOBJ.SKIPCHAR))
elseif (< SKIPPING 0)
then ; in a nice ascii section
(BOUT OUTSTREAM CH)
elseif HELDCH
then ; We were just waiting to see...
(BOUT OUTSTREAM HELDCH)
(SETQ HELDCH NIL)
(SETQ SKIPPING -1)
(BOUT OUTSTREAM CH)
else ; We had been skipping. Don't print this byte until we see the
; next byte is nice, too, so as to reduce the gibberish of
; accidental ascii in the middle of binary
(SETQ HELDCH CH))
OUTSTREAM])
```

(\MAILOBJ.PARSE.ATTRIBUTES

```
[LAMBDA (DATA FIELDS START) ; Edited 14-Feb-90 16:26 by bvm
;; Parse the SUBJECT field out of the serialized stream DATA beginning at START. FIELDS is in the format of \nsmail.attributes entries
(SETFILEPTR DATA (+ START 4)) ; Skip the version number (LONGCARDINAL). Next comes
; SEQUENCE Filing.Attribute
(to (\WIN DATA) bind (CNT _ (LENGTH FIELDS))
X TYPE
do (SETQ TYPE (COURIER.READ DATA NIL 'LONGCARDINAL))
(if (find old X in FIELDS suchthat (EQ (CADR X)
```

```

                                TYPE))
    then
        (\WIN DATA)
        (push $$VAL (CAR X)
         (COURIER.READ DATA NIL (CADDR X)))
        (if (<= (SETQ CNT (SUB1 CNT))
            0)
            then ;; Found them all
                (RETURN $$VAL))
        else (COURIER.SKIP.SEQUENCE DATA NIL 'UNSPECIFIED])
)

(ADDTOVAR FILING.TYPES (VIEWPOINT 4353)
 (RES 4428)
 (XEROX860 5120)
 (REFERENCE 4427)
 (MAILFOLDER 4417))

(RPAQQ MAILOBJ.REFERENCE.FIELD
 (REFERENCE 4421 (NAMEDRECORD (FILE.ID (FILING . FILE.ID))
 (SERVICE NSNAME)
 (ADDRESS NSADDRESS)
 (HOST STRING)
 (DIRECTORY STRING)
 (NAME STRING)
 (TYPE (FILING . ATTRIBUTE.TYPE))
 (NIL UNSPECIFIED)
 (PAGES CARDINAL)
 (VERSION CARDINAL)
 (FLAGS CARDINAL))))))

(RPAQ? MAILOBJ.WINDOWOFFSET 16)

(RPAQ? MAILOBJ.SKIPCHAR (CHARCODE "."))

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD MAILOBJ (MAILOBJ.IMAGE MAILOBJ.BOX MAILOBJ.TYPE MAILOBJ.DATA MAILOBJ.ATTR.LENGTH MAILOBJ.START
 MAILOBJ.NAME MAILOBJ.EXPANDABLE . MAILOBJ.INFO))

(DECLARE%: EVAL@COMPILE

(RPAQQ \MAILOBJ.REFERENCE.LAST.FILED 8192)

(CONSTANTS \MAILOBJ.REFERENCE.LAST.FILED)
)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\MAILOBJ.INIT)

(AND (EQ MAKESYSNAME :LYRIC)
 (FILESLOAD (SYSLOAD)
 NSRANDOM)
)

;; sending mail

(DEFINEQ

(\NSMAIL.SEND.PARSE
 [LAMBDA (MSG EDITORWINDOW)
 (PROG ((SENDER (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of *LAFITE-MODE-DATA*))
 RECIPIENTS MSGFIELDS FORMATTEDP HEADEREOF INTERESTINGFIELDS SUBJECT ATTACHMENT)
 (OR (SETQ MSGFIELDS (\LAFITE.PREPARE.SEND MSG EDITORWINDOW \LAPARSE.NSMAIL))
 (RETURN))
 [COND
 ((EQ (CAAR MSGFIELDS)
 'EOF)
 (SETQ HEADEREOF (CADR (pop MSGFIELDS))
 (for PAIR in MSGFIELDS
 do (SELECTQ (CAR PAIR)
 ((To cc From Reply-to)
 (push INTERESTINGFIELDS (RPLACD PAIR (\NSMAIL.PARSE (CDR PAIR)
 SENDER EDITORWINDOW)))
 (SELECTQ (CAR PAIR)
 ((To cc)
 [LET [(EXPANDED (for NAME in (CDR PAIR)
 join (if (CL:STRING= (fetch NSDOMAIN of NAME)
 ";")
 then

```

```

; DL syntax
(NSMAIL.EXPAND.DL (fetch NSOBJECT of NAME)
  SENDER EDITORWINDOW)
else (LIST NAME]
  (SETQ RECIPIENTS (COND
    (RECIPIENTS (NS.REMOVEDUPLICATES (APPEND EXPANDED
      RECIPIENTS)
    ))
    (T EXPANDED]))
    ; Might want to check validity of From and Reply-to
  (PROGN
    (NIL)))
  ((Subject In-Reply-to)
    [RPLACD PAIR (COND
      ((CDDR PAIR)
        (CONCATLIST (CDR PAIR)))
      (T (CADR PAIR]
        ; Make one string
      (push INTERESTINGFIELDS PAIR)
    [COND
      ((EQ (CAR PAIR)
        'Subject)
        (SETQ SUBJECT (CDR PAIR))
        (Date (\SENDMESSAGEFAIL EDITORWINDOW "User-supplied Date not allowed"))
        (Sender (\SENDMESSAGEFAIL EDITORWINDOW "User-supplied Sender not allowed"))
        (Format (SETQ FORMATTEDP (SELECTQ (CADR PAIR)
          (TEDIT T)
          NIL)))
        ((REFERENCE ATTACHMENT)
          (if ATTACHMENT
            then (\SENDMESSAGEFAIL EDITORWINDOW "Can only send a single attachment"))
          (SETQ ATTACHMENT T)
          (push INTERESTINGFIELDS PAIR))
        NIL))
    (COND
      ((NULL RECIPIENTS)
        (\SENDMESSAGEFAIL EDITORWINDOW "No recipients!"))
      (OR FORMATTEDP (SELECTQ (\LAFITE.CHOOSE.MSG.FORMAT MSG NIL EDITORWINDOW)
        (TEDIT (SETQ FORMATTEDP T))
        NIL)
        ; Aborted
      (RETURN))
    NIL))
  (RETURN (create NSMAILPARSE
    NSPSUBJECT _ SUBJECT
    NSPRECIPIENTS _ RECIPIENTS
    NSPSTART _ HEADEREOF
    NSPFIELDS _ INTERESTINGFIELDS
    NSPFORMATTED _ FORMATTEDP]))

```

(NSMAIL.PARSE.REFERENCE

; Edited 17-Jan-89 15:55 by bvm

```

[LAMBDA (FILENAME EDITWINDOW)
  (LET*
    ((*UPPER-CASE-FILE-NAMES* NIL)
      (FULLNAME (FINDFILE FILENAME T)))
    (COND
      ((NULL FULLNAME)
        (\SENDMESSAGEFAIL EDITWINDOW "Can't find reference file " FILENAME))
      (T
        (LET*
          [(FIELDS (UNPACKFILENAME.STRING FULLNAME))
            (HOST (LISTGET FIELDS 'HOST))
            (NSHOST (PARSE.NSNAME HOST))
            (ADDRESS (LOOKUP.NS.SERVER NSHOST))
            (NAME (LISTGET FIELDS 'NAME))
            (EXT (LISTGET FIELDS 'EXTENSION))
            (VERSION (LISTGET FIELDS 'VERSION))
            (ID (GETFILEINFO FULLNAME 'FILE.ID))
            (TYPE (GETFILEINFO FULLNAME 'FILE.TYPE))
            (SIZE (GETFILEINFO FULLNAME 'SIZE)]
          (COND
            ((NOT (AND (STRPOS ":" HOST)
              ADDRESS))
              (\SENDMESSAGEFAIL EDITWINDOW "Reference file must be on NS server"))
            ((NOT (AND ID TYPE SIZE))
              (\SENDMESSAGEFAIL EDITWINDOW "Can't lookup info on " FULLNAME))
            (T `((FILE.ID ,ID)
              (SERVICE ,NSHOST)
              (ADDRESS ,ADDRESS)
              (HOST ,HOST)
              [DIRECTORY ,(CL:SUBSTITUTE #\ / #\> (UNPACKFILENAME.STRING FULLNAME 'DIRECTORY)
                (NAME ,(if EXT
                  then (SETQ NAME (CONCAT NAME "." EXT))
                  else NAME))
              (TYPE ,(if (OR (NEQ TYPE 0)
                (NULL EXT))
                then
                  ; Interesting type, or no clue from extension
                  TYPE
                elseif [AND (SETQ TYPE (\NSMAIL.GUESS.FILE.TYPE NAME EXT))

```

```

(SELECTQ (\SENDMESSAGE.MENUPROMPT EDITWINDOW
          (\LAFITE.CREATE.MENU `((, (CONCAT "Change file type to "
                                           TYPE)
                                           T)
                                           ("Leave as type BINARY" NIL)
                                           ("Abort" 'ABORT))
          "Fix type of reference file?")
          "Referenced document is of type BINARY; some mail clients
          will not understand.")
(NIL NIL)
(ABORT (ERROR!))
(if (SETFILEINFO FULLNAME 'TYPE (SETQ TYPE (\FILETYPE.FROM.TYPE TYPE)))
    then TYPE
    else (\SENDMESSAGEFAIL EDITWINDOW "Could not set the file type"]
      ; Oh, give up, leave it binary
else
  0))
(NIL 0)
(PAGES , (ADD1 SIZE))
(VERSION , (OR (AND VERSION (MKATOM VERSION))
               0))
(FLAGS 0])

```

(\NSMAIL.EXPAND.DL

```

[LAMBDA (DL SENDER EDITWINDOW) ; Edited 16-Jan-89 14:04 by bvm
  (LET ((FILENAME (PACKFILENAME.STRING 'BODY (if (EQL (CL:CHAR DL 0)
                                                    #\")
          then ; quoted file name, take off the quotes first
              (CL:SUBSEQ DL 1 (- (CL:LENGTH DL)
                                1))
          else DL)
        'EXTENSION LAFITEDL.EXT))
    STREAM)
  (if [NULL (SETQ FILENAME (if (OR (UNPACKFILENAME.STRING FILENAME 'HOST)
                                  (UNPACKFILENAME.STRING FILENAME 'DIRECTORY))
                              then (INFILEP FILENAME)
                              else ; Search default directories
                                  (FINDFILE FILENAME T (CONS LAFITEDEFAULTHOST&DIR LAFITEDLDIRECTORIES])
    then (\SENDMESSAGEFAIL EDITWINDOW "Can't find file named " DL)
    elseif [NULL (SETQ STREAM (CAR (NLSETQ (OPENTEXTSTREAM (MKATOM FILENAME)
    then (\SENDMESSAGEFAIL EDITWINDOW "Can't open " DL)
    else (RESETLST
          (RESETSAVE NIL (LIST 'CLOSEF STREAM)) ; I hope this closes the file. We used OPENTEXTSTREAM
          ; instead of OPEN so that file can contain tedit formatting.
          (bind LINE while (SETQ LINE (CL:READ-LINE STREAM NIL NIL)) join (\NSMAIL.PARSE LINE SENDER
                                                                           EDITWINDOW)))]))

```

(\NSMAIL.PARSE

```

[LAMBDA (FIELD DEFAULTDOMAIN EDITWINDOW) (* bvm%: "3-Jul-84 16:21")
  (\NS.REMOVEDUPLICATES (COND
    ((LISTP FIELD)
     (for PIECE in FIELD join (\NSMAIL.PARSE1 PIECE DEFAULTDOMAIN EDITWINDOW)))
    (T (\NSMAIL.PARSE1 FIELD DEFAULTDOMAIN EDITWINDOW]))

```

(\NSMAIL.PARSE1

```

[LAMBDA (FIELD DEFAULTDOMAIN EDITWINDOW) (* bvm%: "3-Jul-84 16:26")
  (COND
    (FIELD (bind ADDR (START _ 1)
                  COMMA when (PROGN [SETQ ADDR (SUBSTRING FIELD START (COND
                                                                 (SETQ COMMA (STRPOS '%, FIELD START)
                                                                 )
                                                                 (SUB1 COMMA])
    (do ; Strip leading blanks
      (SELCHARQ (CHCON1 ADDR)
                ((SPACE TAB)
                 (GNC ADDR))
                (RETURN)))
    (do ; Strip trailing blanks
      (SELCHARQ (NTHCHARCODE ADDR -1)
                ((SPACE TAB)
                 (GLC ADDR))
                (RETURN)))
    (NEQ (NCHARS ADDR)
         0))
  collect (PARSE.NSNAME ADDR NIL DEFAULTDOMAIN) repeatwhile (COND
    (COMMA (SETQ START (ADD1 COMMA]))

```

(\NS.REMOVEDUPLICATES

```

[LAMBDA (LST) ; Edited 6-Jun-88 13:38 by bvm
  (CL:REMOVE-DUPLICATES LST :TEST (FUNCTION EQUAL.CH.NAMES]))

```

(\NSMAIL.SEND

```

[LAMBDA (MSG PARSE EDITORWINDOW ABORTWINDOW) ; Edited 26-Jun-90 18:25 by jds

```


::: MSG is the entire text of the message -- RECIPIENTS is a parsed list of recipients

```
(DECLARE (SPECVARS MSG START MSGFIELDS EDITORWINDOW ABORTWINDOW FORMATSTREAM REFERENCE ATTACHMENT
ATTACHED-ATTRIBUTES BODYTYPE BODYLENGTH NOTEP)
; For \NSMAIL.SEND.MESSAGE.CONTENT
(RESETLST
(PROG ((PWINDOW (AND EDITORWINDOW (GETPROMPTWINDOW EDITORWINDOW)))
(RECIPIENTS (fetch (NSMAILPARSE NSPRECIPIENTS) of PARSE))
(START (OR (fetch (NSMAILPARSE NSPSTART) of PARSE)
(GETEOFPTR MSG)))
(MSGFIELDS (fetch (NSMAILPARSE NSPFIELDS) of PARSE))
(CREDENTIALS (fetch (LAFITEMODEDATA CREDENTIALS) of *LAFITE-MODE-DATA*))
FORMATSTREAM REFERENCE ATTACHMENT BODYTYPE BODYLENGTH NOTEP COURIERSTREAM DATASTREAM
RECIPIENTSCHECK SENDRESULT SENDERFIELD DATEFIELD TYPE MAILDROP RESULTS ATTACHED-ATTRIBUTES)
[COND
(PWINDOW
; Make sure prompt window will expand as needed. Probably
; generic sendmessage should do this
(RESETSAVE (TTYDISPLAYSTREAM PWINDOW))
(RESETSAVE (LINELENGTH T])
(COND
((AND (fetch (NSMAILPARSE NSPFORMATTED) of PARSE)
(TEDIT.FORMATTEDFILEP MSG))
; Message is formatted, so get info. Have to exclude header,
; since it is not sent.
(SETQ MSG (COPYTEXTSTREAM MSG))
(TEDIT.DELETE MSG 1 START)
(SETQ FORMATSTREAM (COERCETEXTOBJ MSG 'SPLIT)) ; Get (body . formatting)
(SETQ MSG (OPENSTREAM (CAR FORMATSTREAM)
'INPUT))
(SETQ FORMATSTREAM (OPENSTREAM (CDR FORMATSTREAM)
'INPUT))
(SETQ START 0))
((AND (TEXTSTREAMP MSG)
(TEDIT.FORMATTEDFILEP MSG))
; Message has formatting, but caller asked to send it as plain
; text. Carefully coerce it, since TEDIT ns chars and image
; objects don't pass thru COPYBYTES very well
(SETQ MSG (LAFITE.MAKE.PLAIN.TEXTSTREAM MSG START))
(SETQ START 0)))
(SETQ BODYLENGTH (- (GETEOFPTR MSG)
START))
(SETQ REFERENCE (ASSOC 'REFERENCE MSGFIELDS))
(SETQ ATTACHMENT (ASSOC 'ATTACHMENT MSGFIELDS))
(if (OR REFERENCE ATTACHMENT)
then
; Text must be sent as mail note
(if (< BODYLENGTH *NSMAIL-MAX-NOTE-LENGTH*)
then (SETQ NOTEP T)
else (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Message text too long to send with
attachment"))))
(if (AND REFERENCE ATTACHMENT)
then (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Can't send both attachment file AND
reference"))))
elseif (AND *NSMAIL-SEND-MAIL-NOTES* (< BODYLENGTH *NSMAIL-MAX-NOTE-LENGTH*))
then (SETQ NOTEP T))
(if ATTACHMENT
then (SETQ MSGFIELDS (DREMOVE ATTACHMENT MSGFIELDS))
(SETQ ATTACHMENT (\NSMAIL.PREPARE.ATTACHMENT (CADR ATTACHMENT)))
elseif REFERENCE
then (RPLACD REFERENCE (\NSMAIL.PARSE.REFERENCE (CADR REFERENCE)
EDITORWINDOW))
(SETQ BODYTYPE \NSMAIL.REFERENCE.BODYTYPE))
[COND
(PWINDOW (CLEARW PWINDOW)
(LET ((TYPE (if REFERENCE
then (CADR (ASSOC 'TYPE (CDR REFERENCE)))
else BODYTYPE)))
(CL:FORMAT PWINDOW "Delivering ~:[~;formatted ~]~@[with ~A ~]~@[~A ~]to ~D
recipient~:P" FORMATSTREAM [AND TYPE (CL:STRING-CAPITALIZE
(MKSTRING (\TYPE.FROM.FILETYPE TYPE)
(COND
(REFERENCE "reference")
(ATTACHMENT "attachment"))
(LENGTH RECIPIENTS])
[COND
((NULL (SETQ MAILDROP (\NSMAIL.FINDSERVER)))
(RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Couldn't find a mail drop"))
(to 3 until (SETQ COURIERSTREAM (COURIER.OPEN MAILDROP NIL T 'NSMAILER)) do
; loop 3 times trying to start this send
(DISMISS 1000))
[COND
((NULL COURIERSTREAM)
(RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Couldn't connect to a maildrop"))
(RESETSAVE NIL (LIST (FUNCTION \SPP.RESETCLOSE)
COURIERSTREAM))
(AND PWINDOW (printout PWINDOW '|...|))
(SETQ RESULTS (COURIER.CALL COURIERSTREAM 'MAILTRANSPORT 'POST (CAR CREDENTIALS)
(CDR CREDENTIALS)
RECIPIENTS NIL T \NSMAIL.CTSTANDARD.MESSAGE NIL (FUNCTION
```

\NSMAIL.SEND.MESSAGE.CONTENT
)

```

' RETURNERRORS))
[COND
  ((EQ (CAR (LISTP RESULTS))
        ' ERROR)
   (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW (SELECTQ (CADR RESULTS)
                                                     (INVALID.RECIPIENTS
                                                      (\LAFITE.INVALID.RECIPIENTS (CDDR RESULTS)))
                                                     (MKSTRING (CDR RESULTS)
                                                      (AND NSMAILDEBUGFLG (printout PROMPTWINDOW T "Post results: " RESULTS))
                                                      (RETURN (LENGTH RECIPIENTS))))))

```

(\NSMAIL.PREPARE.ATTACHMENT

; Edited 14-Sep-89 12:15 by bvm

```

[LAMBDA (FILE)
  (DECLARE (USEDFREE MSGFIELDS EDITORWINDOW ATTACHMENT ATTACHED-ATTRIBUTES BODYTYPE BODYLENGTH))
  (LET*
    ((HOST (UNPACKFILENAME.STRING FILE 'HOST))
     (SERIALIZED (STRPOS ":" HOST))
     (ATTRCOUNT 0)
     ATTRSTREAM)
    (CL:MULTIPLE-VALUE-BIND (STREAM CONDITION)
      [IGNORE-ERRORS (if SERIALIZED
        then (\NSFILING.GETFILE (\GETDEVICEFROMHOSTNAME (MKATOM (U-CASE HOST)))
                                FILE
                                'SERIALIZE
                                'OLD NIL NIL T)
        else (OPENSTREAM FILE 'INPUT)
      (if (NULL STREAM)
        then (\LAFITE.SEND.FAIL EDITORWINDOW (OR CONDITION "Attachment not found.")
          (ERROR!))
        (RESETSAVE NIL (LIST (FUNCTION CLOSEF)
                            STREAM))
      [if SERIALIZED
        then
          ; Parse out the attributes portion of the serialized file and save
          ; those that are not specifically mail attributes
          (SETQ ATTRSTREAM (OPENSTREAM '{NODIRCORE} 'BOTH))
          (\NSMAIL.CHECK.SERIALIZED.VERSION STREAM)
          [to (\WIN STREAM) bind TYPE WORDCOUNT
            do (SETQ TYPE (COURIER.READ STREAM NIL 'LONGCARDINAL))
              (if (EQ TYPE (\NSMAIL.ATTRIBUTE.TYPE BodyType))
                then
                  ; We always send type explicitly
                  (\WIN STREAM)
                  (SETQ BODYTYPE (COURIER.READ STREAM NIL 'LONGCARDINAL))
                elseif [OR (for TRIPLE in \NSMAIL.ATTRIBUTES thereis (EQ TYPE (CADR TRIPLE)))
                       (AND (< TYPE 100)
                            (for TRIPLE in \NSFILING.ATTRIBUTES when (EQ TYPE (CADR TRIPLE))
                              do
                                ; Only a few filing attributes are interesting. Is.directory appears
                                ; to be vital (the server won't deserialize something with children
                                ; without it)
                                (RETURN (NOT (FMEMB (CAR TRIPLE)
                                                    ' (IS.DIRECTORY CREATED.BY CREATED.ON MODIFIED.BY
                                                       MODIFIED.ON]
                                then
                                  ; A mail attribute or file-specific file attribute, skip it
                                  (COURIER.SKIP.SEQUENCE STREAM NIL 'UNSPECIFIED)
                                else
                                  ; Save it
                                  (add ATTRCOUNT 1)
                                  (COURIER.WRITE ATTRSTREAM TYPE NIL 'LONGCARDINAL)
                                  (\WOUT ATTRSTREAM (SETQ WORDCOUNT (\WIN STREAM)))
                                  (COPYBYTES STREAM ATTRSTREAM (UNFOLD WORDCOUNT BYTESPERWORD]
                                  (SETQ ATTACHED-ATTRIBUTES (CONS ATTRCOUNT ATTRSTREAM))
                                else
                                  ; Not on an NS server, let's investigate the type
                                  (CASE [SETQ BODYTYPE (\FILETYPE.FROM.TYPE (GETFILEINFO STREAM 'TYPE)
                                    (NIL 0)
                                    ; Under specified
                                    (if (SETQ BODYTYPE (\NSMAIL.GUESS.FILE.TYPE (FULLNAME STREAM)))
                                      then (SETQ BODYTYPE (\FILETYPE.FROM.TYPE BODYTYPE))
                                      elseif (\SENDMESSAGE.MENUPROMPT EDITORWINDOW (\LAFITE.CREATE.MENU ' ("Send as BINARY
                                                                                                     attachment" T)
                                                                                                     ("Abort" NIL))
                                                                                                     "Send attachment?")
                                      "Warning: Type of attached file is unknown; most mail clients can't do
                                      anything interesting with this.")
                                    then (SETQ BODYTYPE 0)
                                    else (ERROR!))
                                  (push MSGFIELDS `(MODIFIED.ON ,@(GETFILEINFO STREAM 'ICREATIONDATE]
                                  STREAM)])

```

(\NSMAIL.GUESS.FILE.TYPE

; Edited 17-Jan-89 15:42 by bvm

```

[LAMBDA (FILENAME EXT)
  ;; Given a file name, try to guess what type it is from the extension, since file's TYPE property was boring. EXT is computed from FILENAME if
  ;; omitted.
  (OR (CAR (CL:ASSOC (OR EXT (UNPACKFILENAME.STRING FILENAME 'EXTENSION))
                    FILING.TYPES :TEST 'STRING-EQUAL))
      (LET ((TYPE (PRINTFILETYPE.FROM.EXTENSION FILENAME)))

```

(AND TYPE (CAR (CL:ASSOC TYPE FILING.TYPES :TEST 'STRING-EQUAL]))

(\NSMAIL.SEND.MESSAGE.CONTENT

[LAMBDA (DATASTREAM) ; Edited 13-Sep-89 17:15 by bvm
(DECLARE (USEDFREE MSG START MSGFIELDS EDITORWINDOW ABORTWINDOW FORMATSTREAM REFERENCE ATTACHMENT
ATTACHED-ATTRIBUTES BODYTYPE BODYLENGTH NOTEP)
; From \NSMAIL.SEND

:: Transmits the bulkdata portion of the message

(PROG ((SENDER (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of *LAFITE-MODE-DATA*)))

:: Want to send a serialized file on DATASTREAM --- version plus SerializedTree. See \NSMAIL.READ.SERIALIZED.TREE

(COURIER.WRITE DATASTREAM \SERIALIZED.FILE.VERSION NIL 'LONGCARDINAL)
; Version

:: Now comes (SEQUENCE ATTRIBUTE); the attributes we want to send are those in MSGFIELDS plus Date, From, BodyType and Note

(\WOUT DATASTREAM (+ (LENGTH MSGFIELDS)
(if FORMATSTREAM
then 1 ; Also a LispFormatting item
else 0)
(if NOTEP
then 1 ; Send body as Note attribute
(setq BODYLENGTH 0)
else 1 ; Send as body
0)
(if ATTACHED-ATTRIBUTES
then 1 ; From serialized file
(car ATTACHED-ATTRIBUTES)
else 0)
4)) ; Number of attributes

(\NSMAIL.WRITE.ATTRIBUTE DATASTREAM 'Date (IDATE))
[COND
(ASSOC 'From MSGFIELDS)
(\NSMAIL.WRITE.ATTRIBUTE DATASTREAM 'Sender SENDER)
(T (\NSMAIL.WRITE.ATTRIBUTE DATASTREAM 'From (LIST SENDER)
(for PAIR in MSGFIELDS do (\NSMAIL.WRITE.ATTRIBUTE DATASTREAM (CAR PAIR)
(CDR PAIR)))

(\NSMAIL.WRITE.ATTRIBUTE DATASTREAM 'BodyType (COND
(BODYTYPE)
(NOTEP \NSMAIL.EMPTY.BODYTYPE)
(T \NSMAIL.TEXT.BODYTYPE)))
(\NSMAIL.WRITE.ATTRIBUTE DATASTREAM 'BodySize (if ATTACHMENT
then (setq BODYLENGTH (GETEOPTR ATTACHMENT))
else BODYLENGTH))

(COND
((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW 'ABORT))
(ERROR!)))

[COND
(NOTEP (\NSMAIL.SEND.STREAM.AS.STRING MSG DATASTREAM START (\NSMAIL.ATTRIBUTE.TYPE Note)

[COND
(FORMATSTREAM (\NSMAIL.SEND.STREAM.AS.STRING FORMATSTREAM DATASTREAM 0 (\NSMAIL.ATTRIBUTE.TYPE
LispFormatting]

[PROGN ; Now the content of the serialized tree, first part of which is a
; Bulkdata.StreamOfUnspecified

(COND
(ATTACHMENT (if ATTACHED-ATTRIBUTES
then ; We have a serialized file here already. First send the rest of
; the interesting attributes
(COPYBYTES (CDR ATTACHED-ATTRIBUTES)
DATASTREAM 0 -1) ; Then the rest of the serialization
(COPYBYTES ATTACHMENT DATASTREAM)
else (\COURIER.WRITE.STREAM.UNSPECIFIED DATASTREAM ATTACHMENT 0 BODYLENGTH)))
(NOTEP ; Null content
(\WOUT DATASTREAM 1) ; Last segment
(\WOUT DATASTREAM 0) ; Empty sequence

(T (\COURIER.WRITE.STREAM.UNSPECIFIED DATASTREAM MSG START (GETEOPTR MSG]
(if (NOT ATTACHED-ATTRIBUTES)

then ; Finally, the last of the serialized tree
(\WOUT DATASTREAM (LOGXOR (LOGAND BODYLENGTH 1)
1)) ; Last byte significant (even number of bytes)
(\WOUT DATASTREAM 0) ; No children

(COND
((NULL ABORTWINDOW))
((WINDOWPROP ABORTWINDOW 'ABORT)
(ERROR!))
(T ; Too late to abort now
(DELETEMENU (CAR (WINDOWPROP ABORTWINDOW 'MENU))
NIL ABORTWINDOW)))

(RETURN NIL)]

{COURIER.WRITE.STREAM.UNSPECIFIED

[LAMBDA (OUTSTREAM INSTREAM START END)

(* bvm%: "16-May-85 14:24")

;;; Copies INSTREAM from START to END onto OUTSTREAM in the form of Bulkdata.StreamOfUnspecified --- format is one or more concatenations of
 ;;; {lastSegmentP,SequenceUnspecified} --- returns T if even number of bytes written, NIL if odd

```
(LET (LENGTH)
  [COND
    (END (SETFILEPTR INSTREAM START)
      (SETQ LENGTH (IDIFFERENCE (COND
        ((EQ END -1)
         (GETEOFPTR INSTREAM))
        (T END))
        START)))
    (START (SETQ LENGTH START))
    (T (SETQ LENGTH (IDIFFERENCE (GETEOFPTR INSTREAM)
      (GETFILEPTR INSTREAM)
      (while (GREATERP LENGTH MAX.BULK.SEGMENT.LENGTH) do (\WOUT OUTSTREAM 0)
        ; Not last segment
        (\WOUT OUTSTREAM (FOLDHI MAX.BULK.SEGMENT.LENGTH
          BYTESPERWORD))
        ; Word length of this segment
        (COPYBYTES INSTREAM OUTSTREAM
          MAX.BULK.SEGMENT.LENGTH)
        (SETQ LENGTH (IDIFFERENCE LENGTH
          MAX.BULK.SEGMENT.LENGTH)))
        ; Last segment
        (\WOUT OUTSTREAM 1)
        ; Word length of this segment
        (\WOUT OUTSTREAM (FOLDHI LENGTH BYTESPERWORD))
        (COPYBYTES INSTREAM OUTSTREAM LENGTH)
        (COND
          ((EVENP LENGTH)
           T)
          (T
           ; Garbage last byte
           (\BOUT OUTSTREAM 0)
           NIL]))))])
```

{NSMAIL.SEND.STREAM.AS.STRING

[LAMBDA (INSTREAM OUTSTREAM START ATTRIBUTE)

(* bvm%: "30-Jul-84 15:31")

;; Writes the contents of INSTREAM, beginning at byte START, to OUTSTREAM in the form of a Filing Attribute whose type is ATTRIBUTE and
 ;; whose value is a string

```
(PROG ((EOF (GETEOFPTR INSTREAM))
  LENGTH)
  (COURIER.WRITE OUTSTREAM ATTRIBUTE NIL 'LONGCARDINAL)
  (\WOUT OUTSTREAM (ADD1 (FOLDHI (SETQ LENGTH (IDIFFERENCE EOF START))
    BYTESPERWORD))) ; Sequence length
  (\WOUT OUTSTREAM LENGTH) ; String length
  (COPYBYTES INSTREAM OUTSTREAM START EOF)
  (COND
    ((ODDP LENGTH)
     (\BOUT OUTSTREAM 0]))
```

{NSMAIL.WRITE.ATTRIBUTE

[LAMBDA (STREAM TYPE VALUE)

; Edited 17-Jan-89 16:39 by bvm

```
(LET* [FILINGP (TYPEINFO (if (EQ TYPE 'REFERENCE)
  then
  ; This is handled specially so that we don't read references on
  ; input
  MAILOBJ.REFERENCE.FIELD
  else (OR (ASSOC TYPE \NSMAIL.ATTRIBUTES)
    (SETQ FILINGP (ASSOC TYPE \NSFILING.ATTRIBUTES)]
  (if TYPEINFO
    then (COURIER.WRITE STREAM (CADR TYPEINFO)
      NIL
      'LONGCARDINAL) ; Type code
      (COURIER.WRITE.SEQUENCE.UNSPECIFIED STREAM VALUE (if FILINGP
        then 'FILING
        else 'MAILTRANSPORT)
        (CADDR TYPEINFO))
    else (ERROR "Unknown mail attribute" TYPE])
```

{NSMAIL.FINDSERVER

[LAMBDA NIL

(* bvm%: "14-Nov-84 23:47")

```
(PROG ((NULL.AUTHENTICATOR (CONSTANT (COURIER.CREATE (AUTHENTICATION . CREDENTIALS)
  TYPE _ 'SIMPLE VALUE _ NIL)))
  INFO)
  (RETURN (COND
    [(AND \NSMAIL.SERVER.CACHE (find ADDR in \NSMAIL.SERVER.CACHE
      suchthat (\NSMAIL.CHECKSERVER (COURIER.EXPEDITED.CALL
        ADDR \NSMAIL.SOCKET
        'MAILTRANSPORT
        'SERVER.POLL
        NULL.AUTHENTICATOR
        ' (0)
        ' RETURNERRORS])
```

```

((SETQ INFO (COURIER.BROADCAST.CALL \NSMAIL.SOCKET 'MAILTRANSPORT 'SERVER.POLL
              (LIST NULL.AUTHENTICATOR '(0))
              (FUNCTION \NSMAIL.CHECKSERVER)
              NSMAIL.NET.HINT))
 (push \NSMAIL.SERVER.CACHE INFO)
 INFO])

```

(\NSMAIL.CHECKSERVER

[LAMBDA (POLLRESULT) (* bvm%: " 1-Jul-84 15:15")

;; Checks that the result of a SERVER.POLL is useful. Returns the server's address

```

(COND
  ((AND (FIXP (CAR POLLRESULT))
        (ILESSP (CAR POLLRESULT)
                 10))
   (CAR (CADR POLLRESULT])

```

)

(FILESLOAD LAFITEMAIL)

;; for LAFITE.MAKE.PARSE.TABLE

(RPAQQ **NSMAIL.PARSEFIELDS**

```

(("DATE:" LAFITE.READ.LINE.FOR.TOC Date)
 ("SUBJECT:" LAFITE.READ.LINE.FOR.TOC Subject)
 ("SENDER:" LAFITE.READ.NAME.FIELD Sender)
 ("FROM:" LAFITE.READ.NAME.FIELD From)
 ("REPLY-TO:" LAFITE.READ.NAME.FIELD Reply-to)
 ("IN-REPLY-TO:" LAFITE.READ.LINE.FOR.TOC In-Reply-to)
 ("TO:" LAFITE.READ.NAME.FIELD To)
 ("CC:" LAFITE.READ.NAME.FIELD cc)
 ("FORMAT:" LAFITE.READ.FORMAT)
 ("ATTACHED-REFERENCE:" LAFITE.READ.LINE.FOR.TOC REFERENCE)
 ("ATTACHED-FILE:" LAFITE.READ.LINE.FOR.TOC ATTACHMENT)))

```

(RPAQ **\LAPARSE.NSMAIL** (LAFITE.MAKE.PARSE.TABLE NSMAIL.PARSEFIELDS))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \LAPARSE.NSMAIL)

)

(RPAQ? **\NSMAIL.SERVER.CACHE**)

(RPAQ? **NSMAIL.NET.HINT**)

(RPAQ? ***NSMAIL-MAX-NOTE-LENGTH*** 8000)

(RPAQ? ***NSMAIL-SEND-MAIL-NOTES***)

(RPAQ? ***NSMAIL-CACHE-TIMEOUT*** 14400000)

(RPAQ? **LAFITEDL.EXT** "DL")

(CL:PROCLAIM '(GLOBAL NSMAIL.NET.HINT *NSMAIL-MAX-NOTE-LENGTH* *NSMAIL-SEND-MAIL-NOTES* *NSMAIL-CACHE-TIMEOUT*))

(ADDTOVAR **\SYSTEMCACHEVARS** \NSMAIL.SERVER.CACHE)

(DEFINEQ

(\NSMAIL.MESSAGE.P

[LAMBDA (MSG) ; Edited 6-May-88 13:58 by bvm
(AND (STRPOS ":" (fetch (LAFITEMSG FROM) of MSG))
'?])

(\NSMAIL.MESSAGE.FROM.SELF.P

[LAMBDA (MSG) ; Edited 6-May-88 14:37 by bvm

;; True if message is from current user. Easy in NS case because we always make the From field be exactly our full name

```

(STRING-EQUAL (fetch (LAFITEMSG FROM) of MSG)
 (fetch (LAFITEMODEDATA FULLUSERNAME) of *LAFITE-MODE-DATA*])

```

(\NSMAIL.MAKEANSWERFORM

[LAMBDA (MSGDESCRIPTORS MAILFOLDER) ; Edited 6-Jun-88 14:09 by bvm

```

(LET ((MSGFIELDS (\LAFITE.PARSE.MESSAGE MAILFOLDER (OR (CAR (LISTP MSGDESCRIPTORS))
                                                         MSGDESCRIPTORS)))

```

SUBJECT FROM DATE SENDER REPLYTO TO CC ORIGINALREGISTRY OLDFROM NEWTO) ; get the fields from the file

```

(for PAIR in MSGFIELDS do (SELECTQ (CAR PAIR)
  (Subject (SETQ SUBJECT (CADR PAIR)))
  (Sender (SETQ SENDER (CADR PAIR)))
  (From (SETQ FROM (CADR PAIR)))
  (Date (SETQ DATE (CADR PAIR)))

```

```

(Reply-to (SETQ REPLYTO (CDR PAIR)))
(To (SETQ TO (CDR PAIR)))
(cc (SETQ CC (CDR PAIR)))
NIL) ; first parse the strings into recipients
(COND
 [SENDER ; Sender is a mail address, and has the official registry
 (SETQ ORIGINALREGISTRY (PARSE.NSNAME SENDER))
 (SETQ OLDFROM (AND FROM (\NSMAIL.PARSE FROM ORIGINALREGISTRY)
 [FROM ; Have to parse the From field before we can get its registry
 (SETQ ORIGINALREGISTRY (CAR (SETQ OLDFROM (\NSMAIL.PARSE FROM
 (T (LAB.PROMPTPRINT MAILFOLDER T "Can't reply--no FROM or SENDER field"))
 (SETQ NEWTO (OR (AND REPLYTO (SETQ REPLYTO (\NSMAIL.PARSE REPLYTO ORIGINALREGISTRY))
 OLDFROM))
 (LAFITE.FILL.IN.ANSWER.FORM SUBJECT FROM DATE NEWTO
 (CL:SET-DIFFERENCE [COND
 (REPLYTO ; Only this address, so can only cc to self now
 (LIST (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of *LAFITE-MODE-DATA*
 )))
 (T ; Take everyone who got the original, removing duplicates, of
 ; course.
 (\NS.REMOVEDUPLICATES (APPEND (AND TO (\NSMAIL.PARSE TO
 ORIGINALREGISTRY))
 (AND CC (\NSMAIL.PARSE CC
 ORIGINALREGISTRY]
 NEWTO :TEST (FUNCTION EQUAL.CH.NAMES))
 (FUNCTION \NSMAIL.PRINT.NAMES])
)

```

:: Utility for handling mail attributes

```
(PUTPROPS ENVELOPE.ITEM COURIERDEF (\NS.READ.ENVELOPE.ITEM \NS.WRITE.ENVELOPE.ITEM)
```

(DEFINEQ

(\NS.READ.ENVELOPE.ITEM

```
[LAMBDA (STREAM PROGRAM TYPE) ; Edited 22-Nov-89 12:11 by bvm
```

:: Reads a mailing envelope attribute value pair from STREAM, returning a list of two elements, (TYPE VALUE); if the attribute is not a known attribute, TYPE is an integer and VALUE is a sequence of unspecified

```

(LET* [(TYPE (COURIER.READ STREAM NIL 'LONGCARDINAL))
 (VALUETYPE (\COMPUTED.FORM `(CASE TYPE
 (\,@ [for TRIPLE in \NSMAIL.ENVELOPE.ITEM.TYPES
 collect `((CADR TRIPLE)
 [SETQ TYPE ', (CAR TRIPLE)
 ', (CADDR TRIPLE)]))
 (LIST TYPE (if VALUETYPE
 then (\WIN STREAM) ; Skip sequence count
 (COURIER.READ STREAM PROGRAM VALUETYPE)
 else (COURIER.READ.SEQUENCE STREAM NIL 'UNSPECIFIED))
)

```

(\NS.WRITE.ENVELOPE.ITEM

```
[LAMBDA (STREAM ITEM PROGRAM TYPE) ; Edited 22-Nov-89 12:31 by bvm
```

:: Writes a filing attribute value pair to STREAM. ITEM is a list of two elements (TYPE VALUE)

```

(LET ((TYPE (CAR ITEM))
 (VALUE (CADR ITEM))
 VALUETYPE)
 (COURIER.WRITE STREAM
 [OR (FIXP TYPE)
 (\COMPUTED.FORM `(CASE TYPE
 (\,@ [for TRIPLE in \NSMAIL.ENVELOPE.ITEM.TYPES
 collect `((CAR TRIPLE)
 [SETQ VALUETYPE ', (CADDR TRIPLE)
 ', (CADR TRIPLE)]
 (T (ERROR "Unknown Envelope Item Type" TYPE)))]
 NIL
 'LONGCARDINAL)
 (COND
 (VALUETYPE (COURIER.WRITE.SEQUENCE.UNSPECIFIED STREAM VALUE PROGRAM VALUETYPE))
 (T (COURIER.WRITE.SEQUENCE STREAM VALUE PROGRAM 'UNSPECIFIED])
)
)

```

(RPAQQ \NSMAIL.ENVELOPE.ITEM.TYPES

```

((Postmark 0 POSTMARK)
 (Message-ID 1 MESSAGEID)
 (ContentsType 2 LONGCARDINAL)
 (CONTENTS.SIZE 3 LONGCARDINAL)
 (Originator 4 RNAME)
 (TransportProblem 6 PROBLEM)
 (RETURN.TO.NAME 7 RNAME)
 (Previous-Recipients 8 RNAME.LIST)
 (BodyType 17 LONGCARDINAL)
 (Status 1000 (INBASKET . STATUS))))

```

(DECLARE%: EVAL@COMPILE DOCOPY

(RPAQQ \NSMAIL.ATTRIBUTES
((From 4672 NAME.LIST)
(Date 4673 TIME)
(Reply-to 4674 NAME.LIST)
(To 4676 NAME.LIST)
(cc 4677 NAME.LIST)
(Subject 9 STRING)
(Message-ID 4693 MESSAGEID)
(Sender 4705 NAME)
(BodySize 16 LONGCARDINAL)
(BodyType 17 LONGCARDINAL)
(Note 4687 STRING)
(OldLispFormatting 4910 STRING)
(LispFormatting 4911 STRING)
(In-Reply-to 4690 STRING)))
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

[RECORD NSMAILBOX (NSMAILSTREAM NSMAILENVTAIL NSMAILENVELOPES NSMAILLASTINDEX . NSMAILSTATE)
(ACCESSFNS NSMAILBOX ((NSMAILSESSION (fetch STATESESSION of (fetch NSMAILSTATE of DATUM)))
(NSMAILFIRSTINDEX (fetch STATEFIRSTNEW of (fetch NSMAILSTATE of DATUM))

(RECORD NSMAILSTATE (STATESESSION STATEFIRSTNEW STATEOLDLAST STATENAME STATECREDENTIALS STATEADDRESS
STATELASTERROR STATETIMER))

(RECORD NSMAILPARSE (NSPSUBJECT NSPRECIPIENTS NSPSTART NSPFORMATTED . NSPFIELDS)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \NSMAIL.SOCKET 26)

(RPAQQ \SERIALIZED.FILE.VERSION 2)

(RPAQQ \SERIALIZED.FILE.VERSIONS (2 3))

(RPAQQ \NSMAIL.CTSTANDARD.MESSAGE 0)

(RPAQQ \NSMAIL.TEXT.BODYTYPE 2)

(RPAQQ \NSMAIL.EMPTY.BODYTYPE 4)

(RPAQQ \NSMAIL.REFERENCE.BODYTYPE 4427)

(RPAQQ \MAX.BULK.SEGMENT.LENGTH 32768)

(RPAQQ \NULL.CACHE.VERIFIER (0 0 0 0))

(CONSTANTS \NSMAIL.SOCKET \SERIALIZED.FILE.VERSION \SERIALIZED.FILE.VERSIONS \NSMAIL.CTSTANDARD.MESSAGE
\NSMAIL.TEXT.BODYTYPE \NSMAIL.EMPTY.BODYTYPE \NSMAIL.REFERENCE.BODYTYPE MAX.BULK.SEGMENT.LENGTH
\NULL.CACHE.VERIFIER)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \NSMAIL.ATTRIBUTE.TYPE MACRO [ARGS (COND
((CADR (ASSOC (CAR ARGS)
\NSMAIL.ATTRIBUTES)))
(T (ERROR "Unknown mail attribute" (CAR ARGS))
'IGNOREMACRO])

(PUTPROPS \NSMAIL.WRITE.ATTRIBUTE MACRO [ARGS (LET [(INFO (CDR (ASSOC (CAR (CONSTANTEXPRESIONP (CADR ARGS)))
\NSMAIL.ATTRIBUTES]
(COND
[INFO (LIST '\NSMAIL.WRITE.ATTRIBUTE.MACRO (CAR ARGS)
(CAR INFO)
(CADDR ARGS)
(KWOTE (CADR INFO))
(T 'IGNOREMACRO])

(PUTPROPS \NSMAIL.WRITE.ATTRIBUTE.MACRO MACRO (OPENLAMBDA (STREAM TYPENO VALUE VALUETYPE)
(COURIER.WRITE STREAM TYPENO NIL 'LONGCARDINAL)
(COURIER.WRITE.SEQUENCE.UNSPECIFIED STREAM VALUE
'MAILTRANSPORT VALUETYPE)))

(PUTPROPS \NSMAIL.ATTRIBUTE.TYPE INFO NOEVAL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NSMAIL.NET.HINT \NSMAIL.ENVELOPE.ITEM.TYPES \NSMAIL.ATTRIBUTES \NSMAIL.SERVER.CACHE NSMAILDEBUGFLG

NSWIZARDFLG NSMAIL.LEAVE.ATTACHMENTS \NSMAIL.GOOD.BODYTYPES MAILOBJ.WINDOWOFFSET MAILOBJ.SKIPCHAR
\MAILOBJ.IMAGEFNS MAILOBJ.REFERENCE.FIELD \NSFILING.ATTRIBUTES DEFAULTICONFONT NSPRINT.WATCHERFLG
NSMAIL.HEADER.ORDER FILING.TYPES)

)

(CL:PROCLAIM ' (CL:SPECIAL *RETRIEVAL-ERROR*))

(FILESLOAD (SOURCE)
LAFITEDECLS)

(FILESLOAD (LOADCOMP)
CLEARINGHOUSE)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA** INBASKET.CALL)
)

(PUTPROPS **NSMAIL COPYRIGHT** ("Venue & Xerox Corporation" 1984 1985 1986 1987 1988 1989 1990))

FUNCTION INDEX

COURIER.WRITE.STREAM.UNSPECIFIED	28	\NSMAIL.CHECK	5
INBASKET.CALL	8	\NSMAIL.CHECK.SERIALIZED.VERSION	11
NS.CLOSEMAILBOX	14	\NSMAIL.CHECKSERVER	29
NS.FINDMAILBOXES	5	\NSMAIL.COURIER.OPEN	14
NS.NEXTMESSAGE	8	\NSMAIL.DISCARD.SERIALIZED.CONTENT	11
NS.OPENMAILBOX	5	\NSMAIL.EOF.ON.RETRIEVE	9
NS.POLLNEWMAIL	5	\NSMAIL.ERRORHANDLER	14
NS.REMOVEDUPLICATES	24	\NSMAIL.EXPAND.DL	24
NS.RETRIEVEMESSAGE	8	\NSMAIL.FINDSERVER	28
\MAILOBJ.BUTTONEVENTFN	17	\NSMAIL.FIX.MAILBOX.LOCATIONS	7
\MAILOBJ.COPY.BODY	20	\NSMAIL.GUESS.FILE.TYPE	26
\MAILOBJ.COPY.CHILD	21	\NSMAIL.LOGIN	5
\MAILOBJ.COPY.SEQUENCE	21	\NSMAIL.LOGOFF	14
\MAILOBJ.CREATE	15	\NSMAIL.MAKE.MAILSERVERS	4
\MAILOBJ.DISPLAY	16	\NSMAIL.MAKEANSWERFORM	29
\MAILOBJ.DO.COMMAND	18	\NSMAIL.MESSAGE.FROM.SELF.P	29
\MAILOBJ.EXPAND	20	\NSMAIL.MESSAGE.P	29
\MAILOBJ.EXTRACT.TEXT	21	\NSMAIL.PARSE	24
\MAILOBJ.FB	18	\NSMAIL.PARSE.REFERENCE	23
\MAILOBJ.GET	16	\NSMAIL.PARSE1	24
\MAILOBJ.HARDCOPY	18	\NSMAIL.PREPARE.ATTACHMENT	26
\MAILOBJ.IMAGEBOX	17	\NSMAIL.PRINT.HEADERFIELDS	11
\MAILOBJ.INIT	17	\NSMAIL.PRINT.NAMES	13
\MAILOBJ.MUNGE.NAME	20	\NSMAIL.READ.ENVELOPES	8
\MAILOBJ.NS.TO.LISP.NAME	16	\NSMAIL.READ.SERIALIZED.CONTENT	11
\MAILOBJ.PARSE.ATTRIBUTES	21	\NSMAIL.READ.SERIALIZED.TREE	9
\MAILOBJ.PUT	17	\NSMAIL.READ.STRING.AS.STREAM	11
\MAILOBJ.PUT.FILE	18	\NSMAIL.RETRIEVE	9
\MAILOBJ.TYPE.NAME	16	\NSMAIL.SEND	24
\MAILOBJ.VIEW	19	\NSMAIL.SEND.MESSAGE.CONTENT	27
\NS.READ.ENVELOPE.ITEM	30	\NSMAIL.SEND.PARSE	22
\NS.WRITE.ENVELOPE.ITEM	30	\NSMAIL.SEND.STREAM.AS.STRING	28
\NSMAIL.AUTHENTICATE	4	\NSMAIL.SIGNAL.ERROR	14
\NSMAIL.CHANGE.STATUS	15	\NSMAIL.WRITE.ATTRIBUTE	28

VARIABLE INDEX

NSMAIL-CACHE-TIMEOUT	29	MAILOBJ.SKIPCHAR	22	\LAPARSE.NSMAIL	29
NSMAIL-MAX-NOTE-LENGTH	29	MAILOBJ.WINDOWOFFSET	22	\NSMAIL.ATTRIBUTES	31
NSMAIL-SEND-MAIL-NOTES	29	NSMAIL.HEADER.ORDER	15	\NSMAIL.ENVELOPE.ITEM.TYPES	30
FILING.TYPES	22	NSMAIL.LEAVE.ATTACHMENTS	15	\NSMAIL.GOOD.BODYTYPES	15
LAFITEDL.EXT	29	NSMAIL.NET.HINT	29	\NSMAIL.SERVER.CACHE	29
LAFITEMODELST	5	NSMAIL.PARSEFIELDS	29	\SYSTEMCACHEVARS	29
MAILOBJ.REFERENCE.FIELD	22	NSMAILDEBUGFLG	15		

CONSTANT INDEX

MAX.BULK.SEGMENT.LENGTH	31	\NSMAIL.REFERENCE.BODYTYPE	31	\SERIALIZED.FILE.VERSION	31
\MAILOBJ.REFERENCE.LAST.FILEID	22	\NSMAIL.SOCKET	31	\SERIALIZED.FILE.VERSIONS	31
\NSMAIL.CTSTANDARD.MESSAGE	31	\NSMAIL.TEXT.BODYTYPE	31		
\NSMAIL.EMPTY.BODYTYPE	31	\NULL.CACHE.VERIFIER	31		

RECORD INDEX

MAILOBJ	22	NSMAILBOX	31	NSMAILPARSE	31	NSMAILSTATE	31
---------	----	-----------	----	-------------	----	-------------	----

MACRO INDEX

\NSMAIL.ATTRIBUTE.TYPE	31	\NSMAIL.WRITE.ATTRIBUTE	31	\NSMAIL.WRITE.ATTRIBUTE.MACRO	31
------------------------	----	-------------------------	----	-------------------------------	----

PROPERTY INDEX

ENVELOPE.ITEM	30	\NSMAIL.ATTRIBUTE.TYPE	31
---------------	----	------------------------	----

COURIERPROGRAM INDEX

INBASKET	2	MAILTRANSPORT	2
----------	---	---------------	---