

File created: 24-Feb-2024 11:59:44 {WMEDLEY}<library>lafite>LAFITE-NEWNSMAIL.;2

edit by: rmk

changes to: (VARS LAFITE-NEWNSMAILCOMS NEWNSMAILCOMS)
(FNS \NSMAIL.NEW.CHECK)

previous date: 5-Sep-95 16:43:15 {WMEDLEY}<library>lafite>NEWNSMAIL.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ LAFITE-NEWNSMAILCOMS

```
[ (COURIERPROGRAMS NEW.MAILTRANSPORT NEW.INBASKET)
  (COMS
    (FNS \NS.NEW.READ.ENVELOPE.ITEM \NS.NEW.WRITE.ENVELOPE.ITEM)
    (VARS \NSMAIL.NEW.ENVELOPE.ITEM.TYPES)
    (PROP COURIERDEF NEW.ENVELOPE.ITEM))
  (COMS
    (FNS \NS.READ.HEADING.ATTRIBUTE \NS.WRITE.HEADING.ATTRIBUTE)
    (VARS \NSMAIL.HEADING.ATTRIBUTES)
    (PROP COURIERDEF HEADING.ATTRIBUTE))
  (COMS
    (FNS \NSMAIL.READ.RNAME \NSMAIL.WRITE.RNAME \NSMAIL.RNAME.LENGTH)
    (PROP COURIERDEF NEW.RNAME)
    (FNS RNAME.TO.STRING X400.NAME.TO.STRING EQUAL.RNAMES))
  [COMS
    (FNS \NSMAIL.NEW.SEND.PARSE \NSMAIL.CHECK.ENUMERATION \NSMAIL.NEW.SEND
      \NSMAIL.NEW.INVALID.RECIPIENTS \NSMAIL.BUILD.HEADING \NSMAIL.POST.BODY.PART
      \NSMAIL.NEW.PREPARE.ATTACHMENT \NSMAIL.CHECK.ABORT \NSMAIL.NEW.FINDSERVER
      \NSMAIL.NEW.CHECKSERVER)
    (VARS NSMAIL.PARSEFIELDS (\LAPARSE.NSMAIL (LAFITE.MAKE.PARSE.TABLE NSMAIL.PARSEFIELDS)))
    (INITVARS (*NEWNSMAIL-POST-AS-TEXTFILE* :TEST)
      (*NEWNSMAIL-REPORT-TYPE* 'NON.DELIVERY.ONLY)
      (*NSMAIL-ALLOW-DL-RECIPIENTS* T)
      (*NSMAIL-RETURN-CONTENTS* T)
      (*NSMAIL-MIN-WILLINGNESS* 9)
      (*NSMAIL-TRACE-SERVERS*)
      (*NSMAIL-GENERATE-MESSAGE-ID*)
      (*NSMAIL-DISPLAY-TRANSPORT-ID*)
      (*NSMAIL-DISPLAY-POSTMARK*)
      (*NSMAIL-DISPLAY-ERRORS-TO*)
      (*NSMAIL-CACHE-TIMEOUT* (TIMES 1000 60 60))
      (\NSMAIL.MIN.VP.TYPE 4300)
      (\NSMAIL.MAX.VP.TYPE 5200)
      (\NSMAIL.NEW.SERVER.CACHE))
    [VARS (*NSMAIL-OP-VECTOR* ' (NEWNS.POLLNEWMAIL NEWNS.OPENMAILBOX NEWNS.NEXTMESSAGE
      NEWNS.RETRIEVEMESSAGE NEWNS.CLOSEMAILBOX)
      (GLOBALVARS \NSMAIL.NEW.SERVER.CACHE \NSMAIL.MIN.VP.TYPE \NSMAIL.MAX.VP.TYPE)
      (P (CL:PROCLAIM ' (CL:SPECIAL *NEWNSMAIL-POST-AS-TEXTFILE* *NEWNSMAIL-REPORT-TYPE*
        *NSMAIL-ALLOW-DL-RECIPIENTS* *NSMAIL-RETURN-CONTENTS*
        *NSMAIL-MIN-WILLINGNESS* *NSMAIL-TRACE-SERVERS*
        *NSMAIL-GENERATE-MESSAGE-ID* *NSMAIL-DISPLAY-TRANSPORT-ID*
        *NSMAIL-DISPLAY-POSTMARK* *NSMAIL-DISPLAY-ERRORS-TO*
        *NSMAIL-CACHE-TIMEOUT*]
        (COMS
          (FNS \NSMAIL.NEW.AUTHENTICATE NEWNS.POLLNEWMAIL NEWNS.OPENMAILBOX \NSMAIL.NEW.CHECK
            NEWNS.NEXTMESSAGE NEWNS.RETRIEVEMESSAGE \NSMAIL.READ.BODY.PARTS \NSMAIL.COPY.IA5
            \NSMAIL.COPY.NSTEXTFILE \NSMAIL.READ.HEADING \NSMAIL.PARSE.ANNOTATION \NSMAIL.EMIT.ANNOTATION
            LA.TRIM.WHITESPACE \NSMAIL.READ.FORWARDING \NSMAIL.NEW.PRINT.HEADING \NSMAIL.NEW.PRINT.NAMES
            \NSMAIL.EMIT.FORWARDING \NSMAIL.GDATE \NSMAIL.TRANSLATE.IP.MESSAGEID \NSMAIL.MAYBE.QUOTE
            NULL.NSNAME \NSMAIL.HANDLE.DELIVERY.REPORT \NSMAIL.RECIPIENT.NAME NEW.INBASKET.CALL
            NEWNS.CLOSEMAILBOX \NSMAIL.NEW.LOGOFF)
          (VARS \NSMAIL.GOOD.BODY.PARTS \NSMAIL.DISCARDABLE.BODY.PARTS)
          (GLOBALVARS \NSMAIL.GOOD.BODY.PARTS \NSMAIL.DISCARDABLE.BODY.PARTS)
          (ALISTS (LAFITEMODELST NS))
          (FILES LAFITE-NSMAIL))
          (DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)
            (RECORDS FORWARD)
            (MACROS \NSMAIL.BODY.PART.TYPE)
            (GLOBALVARS \NSMAIL.BODY.PART.TYPES \NSMAIL.HEADING.ATTRIBUTES)
            [P (CL:PROCLAIM ' (CL:SPECIAL *RETRIEVAL-ERROR*)
              (FILES (SOURCE)
                LAFITE-DECLS)
              (FILES (LOADCOMP)
                LAFITE-NSMAIL ATBL)
              ; ATBL has \COMPUTED.FORM macro.
              ; This one we need at run time also
              (CONSTANTS * \NSMAIL.CONTENTS.TYPES)
              DOCOPY
              (VARS \NSMAIL.BODY.PART.TYPES))
            (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
              (NLAML)
              (LAMA NEW.INBASKET.CALL]))
```

(COURIERPROGRAM **NEW.MAILTRANSPORT** (17 5)**TYPES**

```

[ (CREDENTIALS (AUTHENTICATION . CREDENTIALS))
  (VERIFIER (AUTHENTICATION . VERIFIER))
  (SESSION (RECORD (TOKEN (ARRAY 2 UNSPECIFIED))
    (VERIFIER VERIFIER)))
  (ENVELOPE.ITEM.TYPE LONGCARDINAL)
  (ENVELOPE (SEQUENCE NEW.ENVELOPE.ITEM))
  (INVALID.NAME (RECORD (ID CARDINAL)
    (REASON INVALID.REASON)))
  (INVALID.NAME.LIST (SEQUENCE INVALID.NAME))
  (INVALID.REASON (ENUMERATION (NoSuchRecipient 0)
    (NoMailboxForRecipient 1)
    (IllegalName 2)
    (NoDlsAllowed 3)
    (ReportNotAllowed 4)))
  (NAME NSNAME)
  (RNAME NEW.RNAME ; (choice (xns 0 name) (gateway 1 gateway.name))
  )
  (RNAME.LIST (SEQUENCE RNAME))
  [GATEWAY.NAME (RECORD (COUNTRY STRING)
    (ADMIN.DOMAIN STRING)
    (PRIVATE.DOMAIN STRING)
    (ORGANIZATION STRING)
    (ORGANIZATIONAL.UNITS (SEQUENCE STRING))
    (PERSONAL (CHOICE (WHOLE 0 STRING)
      (BROKEN 1 BROKEN.NAME)))
    (GATEWAY.SPECIFIC.INFORMATION (SEQUENCE X400.ATTRIBUTE]
  (BROKEN.NAME (RECORD (GIVEN STRING)
    (INITIALS STRING)
    (FAMILY STRING)
    (GENERATION STRING)))
  (X400.ATTRIBUTE (RECORD (TYPE STRING)
    (VALUE STRING)))
  (REPORT.TYPE (ENUMERATION (NONE 0)
    (NON.DELIVERY.ONLY 1)
    (ALL 2)))
  (RECIPIENT (RECORD (NAME RNAME)
    (RECIPIENT.ID CARDINAL)
    (REPORT REPORT.TYPE)))
  (RECIPIENT.LIST (SEQUENCE RECIPIENT))
  (WILLINGNESS (SEQUENCE WILLINGNESS.METRIC))
  (WILLINGNESS.METRIC CARDINAL)
  (BODY.PART.TYPE LONGCARDINAL)
  (CONTENTS.TYPE LONGCARDINAL)
  (MESSAGEID (ARRAY 5 UNSPECIFIED))
  [POSTING.DATA (RECORD (RECIPIENTS RECIPIENT.LIST)
    (CONTENTS.TYPE CONTENTS.TYPE)
    (CONTENTS.SIZE LONGCARDINAL)
    (BODY.PART.TYPES.SEQUENCE (SEQUENCE BODY.PART.TYPE]
  (POSTMARK (RECORD (POSTED.AT RNAME)
    (TIME TIME)))
  (TOC (SEQUENCE TOC.ITEM))
  (TOC.ITEM (RECORD (TYPE BODY.PART.TYPE)
    (SIZE LONGCARDINAL)))
  [REPORT (RECORD (ORIGINAL.ENVELOPE ENVELOPE)
    [FATE (CHOICE (DELIVERED 0 (ENUMERATION (CONTENTS.TRUNCATED 0)
      (NO.PROBLEM 1)))
      (NOT.DELIVERED 1 (RECORD (REASON NON.DELIVERY.REASON)
        (POSTMARK POSTMARK]
      (REPORT.TYPE (CHOICE (DLMEMBER 0 DLREPORT)
        (OTHER 1 OTHER.REPORT]
    ]
  [DLREPORT (RECORD (DLNAME RNAME)
    (INVALID.RECIPIENTS (SEQUENCE NON.DELIVERED.RECIPIENT]
  [OTHER.REPORT (RECORD (SUCCEEDED (SEQUENCE DELIVERED.RECIPIENT))
    (FAILED (SEQUENCE NON.DELIVERED.RECIPIENT]
  (DELIVERED.RECIPIENT (RECORD (RECIPIENT RECIPIENT)
    (WHEN TIME)))
  (NON.DELIVERED.RECIPIENT (RECORD (RECIPIENT RECIPIENT)
    (REASON NON.DELIVERY.REASON)))
  (NON.DELIVERY.REASON (ENUMERATION (NoSuchRecipient 0)
    (NoMailboxForRecipient 1)
    (IllegalName 2)
    (Timeout 3)
    (ReportNotAllowed 4)
    (MessageTooLong 5)
    (AmbiguousRName 6)
    (IllegalCharacters 7)
    (UnsupportedBodyParts 8)
    (UnsupportedContentsType 9)
    (TransientProblem 10)
    (ContentSyntaxError 11)
    (TooManyRecipients 12)
    (ProtocolViolation 13)
    (X400PragmaticConstraintViolation 14)
    (x400NoBilateralAgreement 15)
    (AccessRightsInsufficientForDL 16)

```

```

        (Other 17)))
(TRANSPORT.OPTIONS (RECORD (RETURN.OF.CONTENTIS BOOLEAN)
        (ALTERNATE.RECIPIENT.ALLOWED BOOLEAN)))
(PRIORITY (ENUMERATION (NonUrgent 0)
        (Normal 1)
        (Urgent 2)))
(CONVERTED.ITEM (ENUMERATION (IA5TextToTeletex 0)
        (TeletexToTelex 1)
        (TeletexToIA5Text 2)
        (TelexToTeletex 3)))
(IP.MESSAGEID (RECORD (ORIGINATOR RNAME)
        (UNIQUESTRING STRING)))
(AUTHENTICATION.LEVEL (ENUMERATION (Strong 0)
        (Simple 1)
        (Foreign 2)))
[FORWARDED.MESSAGE.INFO (RECORD (ENVELOPE ENVELOPE)
        (HEADING (SEQUENCE HEADING.ATTRIBUTE))
        (ASSOCIATED.BODY.PARTS (SEQUENCE BODY.PART.INDEX))
        (INDEX.OF.PARENT.HEADING (CHOICE (NULL 0 (RECORD))
        (NESTED 1 CARDINAL))

(BODY.PART.INDEX CARDINAL)
(SERVICE.PROBLEM (ENUMERATION (CannotAuthenticate 0)
        (ServiceFull 1)
        (ServiceUnavailable 2)
        (MediumFull 3)))
(TRANSFER.PROBLEM (ENUMERATION (Aborted 0)))
(OTHER.PROBLEM (ENUMERATION (Can'tExpedite 0)
        (MalformedMessage 1)
        (IncorrectContentsSize 2)
        (LAST 65535)))
(SESSION.PROBLEM (ENUMERATION (InvalidHandle 0)
        (WrongState 1])

```

PROCEDURES

```

((SERVER.POLL 0 NIL RETURNS (WILLINGNESS (CLEARINGHOUSE . NETWORK.ADDRESS.LIST)
        NAME))
(BEGIN.POST 1 (POSTING.DATA BOOLEAN (SEQUENCE NEW.ENVELOPE.ITEM)
        CREDENTIALS VERIFIER)
        RETURNS
        (SESSION INVALID.NAME.LIST)
        REPORTS
        (AUTHENTICATION.ERROR INVALID.RECIPIENTS SERVICE.ERROR OTHER.ERROR))
(MAILPOLL 7 (NAME CREDENTIALS VERIFIER)
        RETURNS
        (BOOLEAN)
        REPORTS
        (ACCESS.ERROR AUTHENTICATION.ERROR SERVICE.ERROR OTHER.ERROR))
(POST.ONE.BODY.PART 8 (SESSION BODY.PART.TYPE BULK.DATA.SOURCE)
        RETURNS NIL REPORTS (AUTHENTICATION.ERROR SERVICE.ERROR OTHER.ERROR SESSION.ERROR TRANSFER.ERROR))
(END.POST 9 (SESSION BOOLEAN)
        RETURNS
        (MESSAGEID)
        REPORTS
        (AUTHENTICATION.ERROR SERVICE.ERROR OTHER.ERROR SESSION.ERROR TRANSFER.ERROR)))

```

ERRORS

```

((ACCESS.ERROR 0 (ACCESS.PROBLEM))
(AUTHENTICATION.ERROR 1 ((AUTHENTICATION . PROBLEM)))
(INVALID.RECIPIENTS 3 (INVALID.NAME.LIST))
(SERVICE.ERROR 4 (SERVICE.PROBLEM))
(TRANSFER.ERROR 5 (TRANSFER.PROBLEM))
(OTHER.ERROR 6 (OTHER.PROBLEM))
(SESSION.ERROR 7 (SESSION.PROBLEM)))

```

(COURIERPROGRAM **NEW.INBASKET** (18 2)

INHERITS

(NEW.MAILTRANSPORT)

TYPES

```

[(CREDENTIALS (AUTHENTICATION . CREDENTIALS))
(VERIFIER (AUTHENTICATION . VERIFIER))
(NAME NSNAME)
(INDEX LONGCARDINAL)
(RANGE (RECORD (LOW INDEX)
        (HIGH INDEX)))
[MESSAGE.STATUS (RECORD (USER.DEFINED.STATUS CARDINAL)
        (EXISTENCE.OF.MESSAGE (ENUMERATION (NEW 0)
        (KNOWN 1])

(BODY.PART.SEQUENCE (SEQUENCE BODY.PART.INDEX))
(BODY.PART.STATUS (SEQUENCE BOOLEAN))
[BODY.PART.STATUS.CHANGE (RECORD (BODY.PART.INDEX BODY.PART.INDEX)
        (DELETABLE (ENUMERATION (TRUE 0)
        (NOCHANGE 1])

(BODY.PART.STATUS.CHANGE.SEQUENCE (SEQUENCE BODY.PART.STATUS.CHANGE))
(STATUS (RECORD (MESSAGE.STATUS MESSAGE.STATUS)
        (BODY.PART.STATUS BODY.PART.STATUS)))
(SESSION (RECORD (TOKEN (ARRAY 2 UNSPECIFIED))
        (VERIFIER VERIFIER)))
(ANCHOR (ARRAY 5 UNSPECIFIED))
(STATE (RECORD (NEW CARDINAL)

```

```

(TOTAL CARDINAL)))
(WHICH.MESSAGE (ENUMERATION (THIS 0)
(NEXT 1)))
(ACCESS.PROBLEM (ENUMERATION (AccessRightsInsufficient 0)
(AccessRightsIndeterminate 1)
(NoSuchInbasket 2)
(InbasketIndeterminate 3)
(WrongService 4)))
(CONNECTION.PROBLEM (FILING . CONNECTION.PROBLEM))
(SERVICE.PROBLEM (ENUMERATION (CannotAuthenticate 0)
(ServiceFull 1)
(ServiceUnavailable 2)))
(TRANSFER.PROBLEM (ENUMERATION (Aborted 0)))
(SESSION.PROBLEM (ENUMERATION (TokenInvalid 0)))
(OTHER.PROBLEM (ENUMERATION (USE.COURIER 0)
(MalformedMessage 1)
(InvalidOperation 2)
(LAST 65535)))
(INDEX.PROBLEM (ENUMERATION (InvalidIndex 0)
(InvalidBodyPartIndex 1)

```

PROCEDURES

```

((LOGON 5 (NAME CREDENTIALS VERIFIER)
  RETURNS
  (SESSION STATE ANCHOR)
  REPORTS
  (ACCESS.ERROR AUTHENTICATION.ERROR INBASKET.IN.USE SERVICE.ERROR OTHER.ERROR))
(LOGOFF 4 (SESSION)
  RETURNS NIL REPORTS (AUTHENTICATION.ERROR SESSION.ERROR OTHER.ERROR))
(MAILPOLL 7 (NAME CREDENTIALS VERIFIER)
  RETURNS
  (STATE)
  REPORTS
  (ACCESS.ERROR AUTHENTICATION.ERROR SERVICE.ERROR OTHER.ERROR))
(MAILCHECK 6 (SESSION)
  RETURNS
  (STATE)
  REPORTS
  (AUTHENTICATION.ERROR SESSION.ERROR SERVICE.ERROR OTHER.ERROR))
(RETRIEVE.ENVELOPES 2 (INDEX WHICH.MESSAGE SESSION)
  RETURNS
  (ENVELOPE STATUS INDEX))
(RETRIEVE.BODY.PARTS 8 (INDEX BODY.PART.SEQUENCE BULK.DATA.SINK SESSION)
  RETURNS NIL REPORTS (AUTHENTICATION.ERROR INDEX.ERROR SESSION.ERROR SERVICE.ERROR OTHER.ERROR
  TRANSFER.ERROR))
(CHANGE.MESSAGE.STATUS 0 (RANGE BOOLEAN CARDINAL SESSION)
  RETURNS NIL REPORTS (AUTHENTICATION.ERROR INDEX.ERROR SESSION.ERROR SERVICE.ERROR OTHER.ERROR))
(CHANGE.BODY.PARTS.STATUS 3 (INDEX BODY.PART.STATUS.CHANGE.SEQUENCE SESSION)
  RETURNS
  (BOOLEAN)
  REPORTS
  (AUTHENTICATION.ERROR INDEX.ERROR SESSION.ERROR SERVICE.ERROR OTHER.ERROR))
(DELETE 1 (RANGE SESSION)
  RETURNS NIL REPORTS (AUTHENTICATION.ERROR SESSION.ERROR SERVICE.ERROR OTHER.ERROR))
(GET.SIZE 10 (NAME CREDENTIALS VERIFIER)
  RETURNS
  (LONGCARDINAL)
  REPORTS
  (AUTHENTICATION.ERROR ACCESS.ERROR SERVICE.ERROR OTHER.ERROR)))

```

ERRORS

```

((ACCESS.ERROR 0 (ACCESS.PROBLEM))
(AUTHENTICATION.ERROR 1 ((AUTHENTICATION . PROBLEM)))
(SESSION.ERROR 5 (SESSION.PROBLEM))
(SERVICE.ERROR 6 (SERVICE.PROBLEM))
(TRANSFER.ERROR 7 (TRANSFER.PROBLEM))
(OTHER.ERROR 8 (OTHER.PROBLEM))
(INDEX.ERROR 9 (INDEX.PROBLEM))
(INBASKET.IN.USE 10 (NAME)))

```

:: Courier type EnvelopeItem

(DEFINEQ

(NS.NEW.READ.ENVELOPE.ITEM

[LAMBDA (STREAM PROGRAM TYPE)

; Edited 22-Nov-89 12:53 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.NEW.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.NEW.WRITE.ENVELOPE.ITEM

[LAMBDA (STREAM ITEM PROGRAM TYPE)

; Edited 22-Nov-89 12:53 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.NEW.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.NEW.ENVELOPE.ITEM.TYPES

```

((Postmark 0 POSTMARK)
 (Message-ID 1 MESSAGEID)
 (ContentsType 2 CONTENTS.TYPE)
 (TOC 3 TOC)
 (CONTENTS.SIZE 4 LONGCARDINAL)
 (Originator 5 RNAME)
 (REPORT 6 REPORT)
 (RETURN.TO.NAME 7 RNAME)
 (Previous-Recipients 8 RECIPIENT.LIST)
 (GatewayPostmark 9 POSTMARK)
 (AddressChangeNotice 10 RNAME)
 (TRANSPORT.OPTIONS 11 TRANSPORT.OPTIONS)
 (X400SpecificReportInformation 12 (SEQUENCE (SEQUENCE UNSPECIFIED)))
 (OtherRecipients 13 RECIPIENT.LIST)
 (Priority 14 PRIORITY)
 (Converted 15 (SEQUENCE CONVERTED.ITEM))
 (AuthenticationLevelOfSender 16 AUTHENTICATION.LEVEL)))
```

(PUTPROPS **NEW.ENVELOPE.ITEM** **COURIERDEF** (\NS.NEW.READ.ENVELOPE.ITEM \NS.NEW.WRITE.ENVELOPE.ITEM))

;; Courier type HeadingAttribute

(DEFINEQ

(\NS.READ.HEADING.ATTRIBUTE

[LAMBDA (STREAM PROGRAM TYPE)

; Edited 25-Jan-90 16:59 by bvm

;; Reads a mail heading 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

```

(bind (TYPE _ (COURIER.READ STREAM NIL 'LONGCARDINAL)) for X in \NSMAIL.HEADING.ATTRIBUTES
  when (EQ (CADR X)
            TYPE)
  do [RETURN (LIST* (CAR X)
                    (LET* ((RANDP (RANDACCESSP STREAM))
                          (END (+ (UNFOLD (\WIN STREAM)
                                           BYTESPERWORD)
                                (if RANDP
                                    then (GETFILEPTR STREAM)
                                    else 0))))
                      (CONS (COURIER.READ STREAM (OR PROGRAM 'NEW.MAILTRANSPORT)
                            (CADDR X))
                            (if (AND RANDP (NOT (EQL (SETQ HERE (GETFILEPTR STREAM))
                                                       END))))
                          then (if (> HERE END)
                                  then (HELP "Heading attribute overran by " (- HERE END))
                                  else (to (- END HERE) collect (BIN STREAM)
                                           ; TYPE not recognized
                                           (RETURN (LIST TYPE (COURIER.READ.SEQUENCE STREAM NIL 'UNSPECIFIED)))
                                  finally
```

(\NS.WRITE.HEADING.ATTRIBUTE

[LAMBDA (STREAM ITEM PROGRAM TYPE)

; Edited 22-Nov-89 18:17 by bvm

;;; Writes a mail heading attribute value pair to STREAM. ITEM is a list of two elements (TYPE VALUE)

```

(PROG ((TYPE (CAR ITEM))
      (VALUE (CADR ITEM))
      VALUETYPE)
  [COND
```

```

    ((NOT (FIXP TYPE))
     (for X in \NSMAIL.HEADING.ATTRIBUTES when (EQ (CAR X)
                                                    TYPE)
      do (SETQ TYPE (CADR X))
          (SETQ VALUETYPE (CADDR X))
          (RETURN)
      finally (ERROR "Unknown Heading Attribute Type" TYPE]
    (COURIER.WRITE STREAM TYPE NIL 'LONGCARDINAL)
    (COND
     (VALUETYPE (COURIER.WRITE.SEQUENCE.UNSPECIFIED STREAM VALUE PROGRAM VALUETYPE))
     (T (COURIER.WRITE.SEQUENCE STREAM VALUE PROGRAM 'UNSPECIFIED]))
  )

```

```

(RPAQQ \NSMAIL.HEADING.ATTRIBUTES
  (Message-ID 1 IP.MESSAGEID)
  (Sender 2 RNAME)
  (From 3 RNAME.LIST)
  (To 4 RNAME.LIST)
  (cc 5 RNAME.LIST)
  (bcc 6 RNAME.LIST)
  (In-Reply-to 7 IP.MESSAGEID)
  (Obsoletes 8 (SEQUENCE IP.MESSAGEID))
  (References 9 (SEQUENCE IP.MESSAGEID))
  (Subject 10 STRING)
  (Expiration-Date 11 TIME)
  (Reply-By 12 TIME)
  (Reply-to 13 RNAME.LIST)
  (Importance 14 (ENUMERATION (Low 0)
                               (Normal 1)
                               (High 2)))
  (Sensitivity 15 (ENUMERATION (Personal 0)
                               (Private 1)
                               (CompanyConfidential 2)))
  (Auto-Forwarded 16 BOOLEAN)
  (Immutable 17 (RECORD))
  (Reply-Requested-of 18 RNAME.LIST)
  (TextAnnotation 19 STRING)
  (ForwardedHeadings 20 (SEQUENCE FORWARDED.MESSAGE.INFO))
  (newTextAnnotation 199 STRING)
  (BodyOffset 198 LONGCARDINAL)
  (LispFormatting 4911 STRING))

```

```
(PUTPROPS HEADING.ATTRIBUTE COURIERDEF (\NS.READ.HEADING.ATTRIBUTE \NS.WRITE.HEADING.ATTRIBUTE))
```

```
:: Courier type RName
```

```
(DEFINEQ
```

```
(\NSMAIL.READ.RNAME
```

```
[LAMBDA (STREAM PROGRAM TYPE)
```

```
; Edited 29-Nov-89 12:53 by bvm
```

```
:: Special code to read newmailtransport.rname, whose definition is (choice (xns 0 name) (gateway 1 gateway.name)). The xns name we return as
;; an NSNAME object, all other types as if they had been read as the definition reads.
```

```

(LET ((CHOICE (\WIN STREAM)))
  (CASE CHOICE
    (0 (COURIER.READ.NSNAME STREAM PROGRAM 'NSNAME))
    (1 (LIST 'GATEWAY (COURIER.READ STREAM PROGRAM 'GATEWAY.NAME)))
    (T (ERROR "Not a recognized type of RNAME" CHOICE)))))

```

```
(\NSMAIL.WRITE.RNAME
```

```
[LAMBDA (STREAM ITEM PROGRAM TYPE)
```

```
; Edited 29-Nov-89 12:52 by bvm
```

```
:: Special code to write newmailtransport.rname. ITEM can be an NSNAME or a list (GATEWAY gatewayname).
```

```

(if (TYPEP ITEM 'NSNAME)
  then (\WOUT STREAM 0)
      (COURIER.WRITE.NSNAME STREAM ITEM PROGRAM 'NSNAME)
  elseif (EQ (CAR (LISTP ITEM))
            'GATEWAY)
  then (\WOUT STREAM 1)
      (COURIER.WRITE STREAM (CADR ITEM)
                           PROGRAM
                           'GATEWAY.NAME)
  else (ERROR "ARG not RNAME" ITEM])

```

```
(\NSMAIL.RNAME.LENGTH
```

```
[LAMBDA (ITEM PROGRAM TYPE)
```

```
; Edited 29-Nov-89 21:22 by bvm
```

```

(+ 1 (if (TYPEP ITEM 'NSNAME)
  then (COURIER.NSNAME.LENGTH ITEM PROGRAM 'NSNAME)
  elseif (EQ (CAR (LISTP ITEM))
            'GATEWAY)
  then (COURIER.REP.LENGTH (CADR ITEM)
                           PROGRAM
                           'GATEWAY.NAME)
  else (ERROR "ARG not RNAME" ITEM])

```

)

(PUTPROPS **NEW.RNAME COURIERDEF** (\NSMAIL.READ.RNAME \NSMAIL.WRITE.RNAME \NSMAIL.RNAME.LENGTH))

(DEFINEQ

(**RNAME.TO.STRING**

[LAMBDA (NAME FULLFLG)

; Edited 4-Apr-90 17:26 by bvm

(CL:ETYPESCASE NAME

(NSNAME (NSNAME.TO.STRING NAME FULLFLG))

(LIST (**X400.NAME.TO.STRING** NAME)))])(**X400.NAME.TO.STRING**

[LAMBDA (NAME)

; Edited 4-Apr-90 17:27 by bvm

(LET ((SLASH "/"

TMP)

(if (NEQ (CAR NAME)

'GATEWAY)

then (ERROR "ARG NOT X400 NAME" NAME)**else** (SETQ NAME (CADR NAME)))

(CONCATLIST `(. SLASH . . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

COUNTRY of NAME))

(LIST "C=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

ADMIN.DOMAIN of NAME))

(LIST "ADMD=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

PRIVATE.DOMAIN of NAME))

(LIST "PRMD=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

ORGANIZATION of NAME))

(LIST "O=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

ORGANIZATIONAL.UNITS of NAME))

(for UNIT in TMP join (LIST "OU=" UNIT SLASH)))

, . [AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

PERSONAL of NAME))

(CASE (CAR TMP)

(WHOLE (LIST "PN=" (CADR TMP)

SLASH))

(BROKEN [LET ((BROKEN (CADR TMP)))

. BROKEN.NAME)

GIVEN of BROKEN))

(LIST "G=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT

. BROKEN.NAME)

INITIALS of BROKEN))

(LIST "I=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT

. BROKEN.NAME)

FAMILY of BROKEN))

(LIST "S=" TMP SLASH))

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT

. BROKEN.NAME)

GENERATION of BROKEN))

(LIST "GQ=" TMP SLASH)))])

, . (AND (SETQ TMP (COURIER.FETCH (NEW.MAILTRANSPORT . GATEWAY.NAME)

GATEWAY.SPECIFIC.INFORMATION of NAME))

(for PAIR in TMP join (LIST (CAR PAIR)

"="

(CADR PAIR)

SLASH])

(**EQUAL.RNAMES**

[LAMBDA (NAME1 NAME2)

; Edited 4-Apr-90 17:21 by bvm

(if (**type?** NSNAME NAME1)**then** (AND (**type?** NSNAME NAME2)

(EQUAL.CH.NAMES NAME1 NAME2))

else (EQUAL NAME1 NAME2))

)

;; Posting

(DEFINEQ

(**\NSMAIL.NEW.SEND.PARSE**

[LAMBDA (MSG EDITORWINDOW)

; Edited 6-Aug-93 16:40 by bvm

(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)
              (LET [(STR (COND
                ((CDDR PAIR)
                  ; Make one string
                  (CONCATLIST (CDR PAIR)))
                (T (CADR PAIR)
                  [COND
                    ((EQ (CAR PAIR)
                      'Subject)
                      (SETQ SUBJECT STR))
                    (T
                      ; format is different in new protocol
                      (SETQ STR (COURIER.CREATE (NEW.MAILTRANSPORT . IP.MESSAGEID)
                        ORIGINATOR _
                        (create NSNAME
                          NSOBJECT _ ""
                          NSDOMAIN _ ""
                          NSORGANIZATION _ ""
                          UNIQUESTRING _ STR]
                      (RPLACD PAIR STR)
                      (push INTERESTINGFIELDS 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))
                    ((Importance Sensitivity Immutable)
                      (if (AND (> (NCHARS (CADR PAIR))
                        0)
                        (SETQ PAIR (\NSMAIL.CHECK.ENUMERATION PAIR EDITORWINDOW)))
                      then (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.CHECK.ENUMERATION

; Edited 24-Jan-90 16:35 by bvm

```

[LAMBDA (PAIR EDITORWINDOW)
  (LET* ((FIELD (CAR PAIR))
    (VALUE (CADR PAIR))
    (EXPECTED (CADDR (ASSOC FIELD \NSMAIL.HEADING.ATTRIBUTES)))
    FOUND)
    (if (EQ (CAR (LISTP EXPECTED))
      'ENUMERATION)
      then (SETQ EXPECTED (CDR EXPECTED))
      [if (SETQ FOUND (CL:ASSOC VALUE EXPECTED :TEST 'STRING-EQUAL))
        then (CONS FIELD (CAR FOUND))
        else (\SENDMESSAGEFAIL EDITORWINDOW (CL:FORMAT

```



```
[COND
  (PWINDOW
    (RESETSAVE (TTYDISPLAYSTREAM PWINDOW))
    (RESETSAVE (LINELENGTH T))
  )
(COND
  ((AND (fetch NSPFORMATTED of PARSE)
    (TEDIT.FORMATTEDFILE 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)
    ; We're thru with this new textstream, let it clean up after itself.

    (CLOSEF MSG)
    (SETQ MSG (OPENSTREAM (CAR FORMATSTREAM)
      'INPUT))
    (SETQ FORMATSTREAM (OPENSTREAM (CDR FORMATSTREAM)
      'INPUT))

    (SETQ START 0))
  ((AND (TEXTSTREAMP MSG)
    (TEDIT.FORMATTEDFILE 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))

[if FORMATSTREAM
  then
    ; Formatted messages can only go as text files for now, or else
    ; old clients can't receive them

    (SETQ USENSTEXTFILE T)
  else (CASE *NEWNSMAIL-POST-AS-TEXTFILE*
    (NIL)
    ; Always send as note

    )
    ((:TEST)
    ; Send as note only if short enough (the default)

    (if (> BODYLENGTH *NSMAIL-MAX-NOTE-LENGTH*)
      then (SETQ USENSTEXTFILE T)))
    (T (SETQ USENSTEXTFILE T)))
  (SETQ REFERENCE (ASSOC 'REFERENCE MSGFIELDS))
  (SETQ ATTACHMENT (ASSOC 'ATTACHMENT MSGFIELDS))
  (if (OR REFERENCE ATTACHMENT)
    then (if ATTACHMENT
      then
        ; We're going to send a whole file along with the message

        (SETQ MSGFIELDS (DREMOVE ATTACHMENT MSGFIELDS))
        (SETQ ATTACHMENT (\NSMAIL.NEW.PREPARE.ATTACHMENT (CADR ATTACHMENT)
          EDITORWINDOW))
        (SETQ ATTACHMENT-STREAM (CAR ATTACHMENT))
        (SETQ ATTACHED-ATTRIBUTES (CDDR ATTACHMENT))
        (SETQ ATTACHMENT-TYPE (CADR ATTACHMENT))
        [SETQ ATTACHMENT-TYPE (CASE (\TYPE.FROM.FILETYPE ATTACHMENT-TYPE)
          (INTERPRESS (if NIL
            then
```

```

; This way doesn't go thru the backward incompatibility module
; correctly.
(\NSMAIL.BODY.PART.TYPE
  INTERPRESS)
else (\NSMAIL.BODY.PART.TYPE VPDOCUMENT)
)

(DIRECTORY (\NSMAIL.BODY.PART.TYPE VPFOLDER))
(TEXT (\NSMAIL.BODY.PART.TYPE NSTEXTFILE))
(T (if (AND (>= ATTACHMENT-TYPE \NSMAIL.MIN.VP.TYPE)
            (<= ATTACHMENT-TYPE \NSMAIL.MAX.VP.TYPE))
  then
    ; I assume everything in this range is a vpdocument
    (\NSMAIL.BODY.PART.TYPE VPDOCUMENT)
  else (\NSMAIL.BODY.PART.TYPE OTHERNSFILE))))]

elseif REFERENCE
  then
    ; Just a Vp reference. This is a null file with a special attribute
    ; giving the file name, etc
    (SETQ MSGFIELDS (DREMOVE REFERENCE MSGFIELDS))
    [SETQ ATTACHED-ATTRIBUTES (LIST (CONSTANT (CONS 'BodyType
                                                    \NSMAIL.REFERENCE.BODYTYPE))
                                   (RPLACD REFERENCE (\NSMAIL.PARSE.REFERENCE
                                                         (CADR REFERENCE)
                                                         EDITORWINDOW))
                                   (SETQ ATTACHMENT-TYPE (\NSMAIL.BODY.PART.TYPE VPDOCUMENT)))
    (SETQ PART-TYPES (LIST ATTACHMENT-TYPE)))
  (if USENSTEXTFILE
    then (push PART-TYPES (\NSMAIL.BODY.PART.TYPE NSTEXTFILE))
    elseif (> BODYLENGTH 0)
    then (push PART-TYPES (\NSMAIL.BODY.PART.TYPE MULTINATIONAL.NOTE)))
  (SETQ HEADING (\NSMAIL.BUILD.HEADING MSGFIELDS (fetch (LAFITEMODEDATA UNPACKEDUSERNAME)
                                                         of *LAFITE-MODE-DATA*)))
  [SETQ RECIPIENTS (for R in RECIPIENTS as I from 1
                    collect (COURIER.CREATE (NEW.MAILTRANSPORT . RECIPIENT)
                                             NAME _ R RECIPIENT.ID _ I REPORT _ (OR *NEWSMAIL-REPORT-TYPE*
                                                                                       'NON.DELIVERY.ONLY))
  [COND
    (PWINDOW (CLEARW PWINDOW)
      (LET [(TYPE (if REFERENCE
                     then [\TYPE.FROM.FILETYPE (CADR (ASSOC 'TYPE (CDR REFERENCE)
                     elseif ATTACHMENT-TYPE
                     then (for PAIR in \NSMAIL.BODY.PART.TYPES
                           when (EQL ATTACHMENT-TYPE (CADR PAIR))
                           do (RETURN (CAR PAIR)
      (CL:FORMAT PWINDOW "Delivering ~:[~;formatted ~]~@[with ~A ~]~@[~A ~]to ~D
        recipient~:P" FORMATSTREAM (AND TYPE (CL:STRING-CAPITALIZE (MKSTRING TYPE))
        )
      (COND
        (REFERENCE "reference")
        (ATTACHMENT "attachment"))
      (LENGTH RECIPIENTS]
  [SETQ ESTIMATED-SIZE (PROGN
    ;; @###!@ protocol demands that you tell the size of the message almost exactly. Specifically, size
    ;; estimate must not be too large (!), and not be more than 5000 bytes too small. That almost means
    ;; you have to buffer the whole message before you start. We are lazy here and hope that
    ;; serialization overhead and file server size estimates don't screw us up.
    (+ (GETEOFPTR HEADING)
       BODYLENGTH
       (if FORMATSTREAM
         then
           ; This plus a few more bytes of serialized file encoding
           (GETEOFPTR FORMATSTREAM)
         else 0)
       (if ATTACHMENT-STREAM
         then
           ; This is an underestimate in the case of a non-ns file, but that's
           ; the safe direction, and it's only by the small length of the
           ; attributes
           (GETEOFPTR ATTACHMENT-STREAM)
         else 0])
  [COND
    ((NULL (SETQ MAILDROP (\NSMAIL.NEW.FINDSERVER ESTIMATED-SIZE)))
     (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 'NEW.MAILTRANSPORT 'BEGIN.POST
      (COURIER.CREATE (NEW.MAILTRANSPORT . POSTING.DATA)
        RECIPIENTS _ RECIPIENTS CONTENTS.TYPE _ \CT.STANDARD.MESSAGE
        CONTENTS.SIZE _ ESTIMATED-SIZE BODY.PART.TYPES.SEQUENCE _
        (CONS (\NSMAIL.BODY.PART.TYPE HEADING)
              PART-TYPES))
        NIL ALLOW.DL.RECIPIENTS [AND *NSMAIL-RETURN-CONTENTS*

```

```

' ((TRANSPORT.OPTIONS (T T)
(CAR CREDENTIALS)
(CDR CREDENTIALS)
'RETURNERRORS))
[COND
  ((EQ (CAR (LISTP RESULTS))
    'ERROR)
    (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW (SELECTQ (CADR RESULTS)
      (INVALID.RECIPIENTS
        (\NSMAIL.NEW.INVALID.RECIPIENTS (CADDR
          RESULTS
            )
              RECIPIENTS))
        (MKSTRING (CDR RESULTS))

;; RESULTS = (session invalid-recipients)
(SETQ SESSION (CAR RESULTS))
[if (SETQ RESULTS (CADR RESULTS))
  then
    ; Some were invalid. I think we don't get any here because we
    ; didn't say to post anyway.
    (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW (\NSMAIL.NEW.INVALID.RECIPIENTS RESULTS RECIPIENTS
      ]

;; Now post body parts
(\NSMAIL.CHECK.ABORT ABORTWINDOW COURIERSTREAM SESSION)
(\NSMAIL.POST.BODY.PART COURIERSTREAM SESSION (\NSMAIL.BODY.PART.TYPE HEADING)
  HEADING 0 EDITORWINDOW)
(\NSMAIL.CHECK.ABORT ABORTWINDOW COURIERSTREAM SESSION)
[if USENSTEXTFILE
  then (\NSMAIL.POST.BODY.PART COURIERSTREAM SESSION (\NSMAIL.BODY.PART.TYPE NSTEXTFILE)
    MSG START EDITORWINDOW `((BodyType ,@\NSMAIL.TEXT.BODYTYPE)
      ,@(AND FORMATSTREAM `(LispFormatting ,@FORMATSTREAM]
  elseif (> BODYLENGTH 0)
    then (\NSMAIL.POST.BODY.PART COURIERSTREAM SESSION (\NSMAIL.BODY.PART.TYPE MULTINATIONAL.NOTE
      )
        MSG START EDITORWINDOW))
(\NSMAIL.CHECK.ABORT ABORTWINDOW COURIERSTREAM SESSION)
[if ATTACHMENT-TYPE
  then (\NSMAIL.POST.BODY.PART COURIERSTREAM SESSION ATTACHMENT-TYPE ATTACHMENT-STREAM NIL
    EDITORWINDOW ATTACHED-ATTRIBUTES)
    (\NSMAIL.CHECK.ABORT ABORTWINDOW COURIERSTREAM SESSION))
[if ABORTWINDOW
  then
    ; Too late to abort now
    (DELETEMENU (CAR (WINDOWPROP ABORTWINDOW 'MENU))
      NIL ABORTWINDOW))
(SETQ RESULTS (COURIER.CALL COURIERSTREAM 'NEW.MAILTRANSPORT 'END.POST SESSION NIL 'RETURNERRORS)
  )
[if (EQ (CAR (LISTP RESULTS))
  'ERROR)
  then (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW (MKSTRING (CDR RESULTS)
    (AND NSMAILDEBUGFLG (printout PROMPTWINDOW T "EndPost results: " RESULTS))
    (RETURN (LENGTH RECIPIENTS))))))

```

(\NSMAIL.NEW.INVALID.RECIPIENTS

[LAMBDA (INVALID.NAME.LIST RECIPIENTS)

; Edited 19-Dec-89 13:00 by bvm

;; INVALID.NAME.LIST = Sequence (id reason). id is 1-based.

```

[if (CDR INVALID.NAME.LIST)
  then (CONCAT "Invalid recipients: " (SUBSTRING (for PAIR in INVALID.NAME.LIST
    collect (LIST [COURIER.FETCH (NEW.MAILTRANSPORT
      . RECIPIENT)
        NAME of (CAR (NTH RECIPIENTS
          (CAR PAIR]
            (CADR PAIR)))
      2 -2))
    else (DESTRUCTURING-BIND (ID REASON)
      (CAR INVALID.NAME.LIST)
      (CONCAT (COURIER.FETCH (NEW.MAILTRANSPORT . RECIPIENT)
        NAME of (CAR (NTH RECIPIENTS ID)))
        " -- " REASON]))

```

(\NSMAIL.BUILD.HEADING

[LAMBDA (MSGFIELDS SENDER)

; Edited 11-Jul-90 18:03 by bvm

;; Build a heading body part, which is a sequence of attribute. Return a stream

```

[LET ((S (OPENSTREAM "{nodircore}" 'BOTH))
  (COUNT 2))
  (SETFILEPTR S 2)
  [COND
    ((ASSOC 'From MSGFIELDS)
      (\NS.WRITE.HEADING.ATTRIBUTE S (LIST 'Sender SENDER)
        'NEW.MAILTRANSPORT))
    (T
      (\NS.WRITE.HEADING.ATTRIBUTE S (LIST 'From (LIST SENDER))
        'NEW.MAILTRANSPORT]
    ; Save space for the sequence count
    ; Identify actual sender (single name here)
    ; Identify sender as the sole "From" name

```

```

(for PAIR in MSGFIELDS do (\NS.WRITE.HEADING.ATTRIBUTE S (LIST (CAR PAIR)
                                                                (CDR PAIR))
                        'NEW.MAILTRANSPORT)
  (add COUNT 1))
(\NS.WRITE.HEADING.ATTRIBUTE S (LIST 'TextAnnotation (CONCAT "Date: " (DATE (DATEFORMAT TIME.ZONE
                                                                SPACES DAY.OF.WEEK))
                                                                LAFITEEOL))
  'NEW.MAILTRANSPORT) ; Send the Date with time zone, as Cedar does
(if *NSMAIL-GENERATE-MESSAGE-ID*
  then (\NS.WRITE.HEADING.ATTRIBUTE S [LIST 'Message-ID (COURIER.CREATE (NEW.MAILTRANSPORT
                                                                . IP.MESSAGEID)
                                                                ORIGINATOR _ (fetch UNPACKEDUSERNAME
                                                                of *LAFITE-MODE-DATA*)
                                                                UNIQUESTRING _ (DATE (DATEFORMAT
                                                                TIME.ZONE]
                                                                'NEW.MAILTRANSPORT)
    (add COUNT 1))
  (SETFILEPTR S 0)
  (\WOUT S COUNT)
  S])

```

(\NSMAIL.POST.BODY.PART

```

[LAMBDA (COURIERSTREAM SESSION TYPE PARTSTREAM START EDITORWINDOW ATTRIBUTES)
  ; Edited 8-Mar-90 12:14 by bvm
  (LET [(RESULTS (COURIER.CALL COURIERSTREAM 'NEW.MAILTRANSPORT 'POST.ONE.BODY.PART SESSION TYPE
    [FUNCTION (LAMBDA (BULKSTREAM)
      (if ATTRIBUTES
        then ; Create a serialized file on the fly
          (COURIER.WRITE BULKSTREAM \SERIALIZED.FILE.VERSION NIL
            'LONGCARDINAL)
          ; version. Next comes Sequence Attribute
          (\WOUT BULKSTREAM (LENGTH ATTRIBUTES))
          [for PAIR in ATTRIBUTES
            do (if (EQ (CAR PAIR)
              'LispFormatting)
                then ; Do this special so we don't have to cons an enormous string
                  (\NSMAIL.SEND.STREAM.AS.STRING (CDR PAIR)
                    BULKSTREAM 0 (\NSMAIL.ATTRIBUTE.TYPE
                      LispFormatting))
                else (\NSMAIL.WRITE.ATTRIBUTE BULKSTREAM (CAR PAIR)
                  (CDR PAIR)
                  ;; Next comes StreamOfUnspecified, then lastBytelsSignificant
                  (if PARTSTREAM
                    then (COURIER.WRITE BULKSTREAM (
                      COURIER.WRITE.STREAM.UNSPECIFIED
                      BULKSTREAM PARTSTREAM
                      (OR START 0)
                      -1)
                      NIL
                      'BOOLEAN)
                    else ; no content
                      (\WOUT BULKSTREAM 1)
                      ; Last segment
                      (\WOUT BULKSTREAM 0)
                      ; Empty sequence
                      (\WOUT BULKSTREAM 1)
                      ; Last Byte is Significant = Byte Length is Even.
                      )
                      (\WOUT BULKSTREAM 0)
                      ; no children
                      ; PARTSTREAM is already in proper format, just send it
                    (if START
                      then (SETFILEPTR PARTSTREAM START))
                      (COPYBYTES PARTSTREAM BULKSTREAM))
                      ; return NIL so caller can see return value
                      NIL]
                      'RETURNERRORS]
                    (if (EQ (CAR RESULTS)
                      'ERROR)
                      then (COURIER.CALL COURIERSTREAM 'NEW.MAILTRANSPORT 'END.POST SESSION T 'RETURNERRORS)
                      ; Abort the post
                      (\LAFITE.SEND.FAIL EDITORWINDOW (CL:FORMAT NIL "Failed to post ~A because: ~A"
                        (CAR (find TYP in \NSMAIL.BODY.PART.TYPES
                          suchthat (EQ (CADR TYP)
                            TYPE)))
                        (CDR RESULTS)))
                      (ERROR!]))

```

(\NSMAIL.NEW.PREPARE.ATTACHMENT

```

[LAMBDA (FILE EDITORWINDOW) ; Edited 11-Nov-92 17:07 by bvm
  ;; Returns (stream type . attributes)
  (LET* ((HOST (UNPACKFILENAME.STRING FILE 'HOST))
    (SERIALIZED (STRPOS ":" HOST))

```

```

BODYTYPE)
(CL:MULTIPLE-VALUE-BIND (STREAM CONDITION)
 [IGNORE-ERRORS (if SERIALIZED
                    then (\NSMAIL.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!))
[if SERIALIZED
    then ;; Have to copy to core in order to get the length remotely correct. Stupid protocol
        (LET [(COREFILE (OPENSTREAM ' {NODIRCORE} ' BOTH]
              (COPYBYTES STREAM COREFILE)
              (PROG1 (LIST (OPENSTREAM (CLOSEF COREFILE)
                                  ' INPUT)
                          (GETFILEINFO STREAM ' FILETYPE))
                    (CLOSEF STREAM)))
        else ; Not on an NS server, let's investigate the type
            (RESETSAVE NIL (LIST (FUNCTION CLOSEF)
                                STREAM))
            (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!)))
            (LIST* STREAM BODYTYPE `((BodyType ,@BODYTYPE)
                                     (MODIFIED.ON ,@(GETFILEINFO STREAM ' ICREATIONDATE)]))

```

(\NSMAIL.CHECK.ABORT

```

[LAMBDA (ABORTWINDOW COURIERSTREAM SESSION) ; Edited 28-Nov-89 15:06 by bvm
;; Abort a post if user has pressed Abort
(COND
  ((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW ' ABORT)) ; Abort the post
   (COURIER.CALL COURIERSTREAM 'NEW.MAILTRANSPORT 'END.POST SESSION T 'RETURNERRORS)
   (ERROR!))

```

(\NSMAIL.NEW.FINDSERVER

```

[LAMBDA (ESTIMATED-SIZE) ; Edited 25-Jun-90 16:02 by bvm
  (PROG (INFO)
    [if [AND (CDR \NSMAIL.NEW.SERVER.CACHE)
            (NOT (TIMEREXPIRED? (CAR \NSMAIL.NEW.SERVER.CACHE)
                                ESTIMATED-SIZE))]
        then (if (SETQ INFO (find ADDR in (CDR \NSMAIL.NEW.SERVER.CACHE)
                                suchthat (\NSMAIL.NEW.CHECKSERVER (COURIER.EXPEDITED.CALL ADDR
                                                                    \NSMAIL.SOCKET 'NEW.MAILTRANSPORT
                                                                    'SERVER.POLL
                                                                    'RETURNERRORS)
                                ESTIMATED-SIZE)))
                then (RETURN INFO))
    else ;; Cache nonexistent or timed out, so refigure from scratch. We like to time out the cache periodically so that we don't permanently
          ;; latch on to some distant server when local ones are flaky.
            (SETQ \NSMAIL.NEW.SERVER.CACHE (LIST (SETUPTIMER *NSMAIL-CACHE-TIMEOUT* (CAR
                                                                    \NSMAIL.NEW.SERVER.CACHE)
                                                                    )
            )
            ;; Ask around for a server
            (COND
              ((SETQ INFO (COURIER.BROADCAST.CALL \NSMAIL.SOCKET 'NEW.MAILTRANSPORT 'SERVER.POLL NIL
              [FUNCTION (LAMBDA (RESULT)
                (\NSMAIL.NEW.CHECKSERVER RESULT ESTIMATED-SIZE)
                NSMAIL.NET.HINT))
              (push (CDR \NSMAIL.NEW.SERVER.CACHE)
                   INFO)))
              (RETURN INFO))

```

(\NSMAIL.NEW.CHECKSERVER

```

[LAMBDA (POLLRESULT ESTIMATED-SIZE) ; Edited 4-Jun-92 11:31 by bvm
;; Checks that the result of a SERVER.POLL is useful for sending a message of size ESTIMATED-SIZE. Returns the server's address
;; POLLRESULT = (willingness network.address.list name)
(LET ((WILLINGNESS (CAR POLLRESULT))
      (SIZE (OR ESTIMATED-SIZE 4000)))
  ; The i'th element of willingness defines the server's willingness
  ; to accept messages up to size 8^i.

```

```

    (if (AND (LISTP WILLINGNESS)
              (for W in WILLINGNESS as (I _ 8) by (LLSH I 3) while (> I SIZE) always (>= W
                                                                                       *NSMAIL-MIN-WILLINGNESS*
                                                                                       )))
      then (PROG [(BESTADDRESS (CAR (SORT.NSADDRESSES.BY.DISTANCE (CADDR POLLRESULT)
                                                                    (SELECTQ *NSMAIL-TRACE-SERVERS*
                                                                    (NIL NIL)
                                                                    (:ASK (if [NOT (EQ 'Y (RESETFORM (TTY.PROCESS (THIS.PROCESS))
                                                                    (ASKUSER 30 'Y (LIST "Use posting server" (CADDR
                                                                                       POLLRESULT
                                                                                       )
                                                                                       (LIST BESTADDRESS))
                                                                    NIL T]
                                                                    then (RETURN NIL)))
                                                                    (PRINTOUT PROMPTWINDOW T "Using posting server " (CADDR POLLRESULT)
                                                                    " = " BESTADDRESS))
                                                                    (RETURN BESTADDRESS]))
    )

```

(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)
 ("Importance:" LAFITE.READ.LINE.FOR.TOC Importance)
 ("Sensitivity:" LAFITE.READ.LINE.FOR.TOC Sensitivity)
 ("Immutable:" LAFITE.READ.LINE.FOR.TOC Immutable)))

```

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

(RPAQ? ***NEWSMAIL-POST-AS-TEXTFILE*** :TEST)(RPAQ? ***NEWSMAIL-REPORT-TYPE*** 'NON.DELIVERY.ONLY)(RPAQ? ***NSMAIL-ALLOW-DL-RECIPIENTS*** T)(RPAQ? ***NSMAIL-RETURN-CONTENTS*** T)(RPAQ? ***NSMAIL-MIN-WILLINGNESS*** 9)(RPAQ? ***NSMAIL-TRACE-SERVERS***)(RPAQ? ***NSMAIL-GENERATE-MESSAGE-ID***)(RPAQ? ***NSMAIL-DISPLAY-TRANSPORT-ID***)(RPAQ? ***NSMAIL-DISPLAY-POSTMARK***)(RPAQ? ***NSMAIL-DISPLAY-ERRORS-TO***)(RPAQ? ***NSMAIL-CACHE-TIMEOUT*** (TIMES 1000 60 60))

(RPAQ? \NSMAIL.MIN.VP.TYPE 4300)

(RPAQ? \NSMAIL.MAX.VP.TYPE 5200)

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

(RPAQQ ***NSMAIL-OP-VECTOR*** (NEWSNS.POLLNEWMAIL NEWSNS.OPENMAILBOX NEWSNS.NEXTMESSAGE NEWSNS.RETRIEVEMESSAGE
NEWSNS.CLOSEMAILBOX))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \NSMAIL.NEW.SERVER.CACHE \NSMAIL.MIN.VP.TYPE \NSMAIL.MAX.VP.TYPE)

)

```

(CL:PROCLAIM ' (CL:SPECIAL *NEWSMAIL-POST-AS-TEXTFILE* *NEWSMAIL-REPORT-TYPE* *NSMAIL-ALLOW-DL-RECIPIENTS*
                          *NSMAIL-RETURN-CONTENTS* *NSMAIL-MIN-WILLINGNESS* *NSMAIL-TRACE-SERVERS*
                          *NSMAIL-GENERATE-MESSAGE-ID* *NSMAIL-DISPLAY-TRANSPORT-ID* *NSMAIL-DISPLAY-POSTMARK*
                          *NSMAIL-DISPLAY-ERRORS-TO* *NSMAIL-CACHE-TIMEOUT*))

```

;; Retrieving

(DEFINEQ

(\NSMAIL.NEW.AUTHENTICATE

[LAMBDA NIL

; Edited 6-Aug-93 15:08 by bvm

(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)))
      'NONE)
    (T (NS.AUTHENTICATE (SETQ CREDENTIALS (NS.MAKE.SIMPLE.CREDENTIALS
      (CONS FULLNAME (CDR INFO)
      T)
      (printout PROMPTWINDOW T "Cannot authenticate user " (RNAME.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)
          ))
        (NONE "No such user")
        AUTHENTICATED?)
      ".")
      NIL)
    (T (create LAFITEMODEDATA
      FULLUSERNAME _ (RNAME.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])

```

(NEWS.POLLNEWMAIL

```

[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER) ; Edited 18-Dec-89 18:59 by bvm
  (LET ((RESULT (\NSMAIL.NEW.CHECK ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER)))
    (COND
      ((OR (NOT RESULT)
        (EQ (CAR RESULT)
          'ERROR)) ; Server down
      '?)
      ((NEQ RESULT 0)
      RESULT])

```

(NEWS.OPENMAILBOX

```

[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER) ; Edited 18-Dec-89 18:59 by bvm
  (LET ((STREAM (\NSMAIL.COURIER.OPEN ADDRESS))
    NSMAILSTATE N)
    (COND
      ((NULL STREAM)
      NIL)
      ((OR (NULL (SETQ N (\NSMAIL.NEW.CHECK ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER STREAM T)))
        (EQ (CAR N)
          'ERROR)) ; Return error msg
      (CLOSEF STREAM)
      (CONS NIL (CDR N)))
      ((EQ (PROGN (SETQ NSMAILSTATE (fetch MAILSTATE of MAILSERVER))
      N)
      0)
      (\NSMAIL.NEW.LOGOFF NSMAILSTATE STREAM)
      'EMPTY)
      (T ; Return (MAILBOX . properties)
      (CONS (create NSMAILBOX
        NSMAILSTREAM _ STREAM
        NSMAILLASTINDEX _ 0
        NSMAILSTATE _ NSMAILSTATE)
      (LIST ' %OFMESSAGES N])

```

(\NSMAIL.NEW.CHECK

```

[LAMBDA (ADDRESS REGISTEREDNAME CREDENTIALS MAILSERVER STREAM RETURNERRORS)
  ; Edited 24-Feb-2024 11:54 by rmk
  ; Edited 6-Aug-93 16:41 by bvm

```

;;; 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 TIMER)
    (SETQ SESSION (fetch STATESESSION of STATE))
  RETRY

```

```

[COND
  [(NULL SESSION)
   (if JUSTCHECKING
    then
      (SETQ POLLRESULT (COURIER.EXPEDITED.CALL ADDRESS \NSMAIL.SOCKET 'NEW.INBASKET
                        'MAILPOLL
                        (fetch STATENAME of STATE)
                        (CAR CREDENTIALS)
                        (CDR CREDENTIALS)
                        'RETURNERRORS))
      (GO GOTRESULT))
   (COND
    ((NULL STREAM)
     (COND
      ((SETQ STREAM (COURIER.OPEN ADDRESS NIL T 'NSMAIL))
       (RESETSAVE NIL (LIST 'CLOSEF STREAM)))
      (T (RETURN NIL))
     ))
    (COND
     ((EQ [CAR (SETQ SESSION (COURIER.CALL STREAM 'NEW.INBASKET 'LOGON (fetch STATENAME
                                                                of STATE)
                                                                (CAR CREDENTIALS)
                                                                (CDR CREDENTIALS)
                                                                'RETURNERRORS)]
          'ERROR)
      (GO ERROR)))
     (SETQ POLLRESULT (CADR SESSION))
     (replace STATESESSION of STATE with (SETQ SESSION (CAR SESSION))
      (T (SETQ POLLRESULT (COND
                           ((NULL STREAM)
                            (COURIER.EXPEDITED.CALL ADDRESS \NSMAIL.SOCKET 'NEW.INBASKET
                              'MAILCHECK SESSION 'RETURNERRORS))
                           (T (COURIER.CALL STREAM 'NEW.INBASKET 'MAILCHECK SESSION 'RETURNERRORS)]
                           ))
        ; Just checking
        ; result = (session state anchor)
        ; Failed somehow
        ; Session timed out, start a new one
        (replace STATESESSION of STATE with (SETQ SESSION NIL))
        (replace STATEFIRSTNEW of STATE with NIL)
        (replace STATEOLDLAST of STATE with NIL)
        (GO RETRY))
        (T (SETQ SESSION POLLRESULT)
         (GO ERROR))
        (replace STATELASTERROR of STATE with NIL)
        (replace (MAILSERVER CONTINUANCE) of MAILSERVER with NIL)
        (RETURN (COURIER.FETCH (NEW.INBASKET . STATE)
                              TOTAL of POLLRESULT))
        ERROR
        [if [AND [NOT (EQUAL (CDR SESSION)
                              ' (CONNECTION.PROBLEM NoResponse]
          (NOT (EQUAL (CDR SESSION)
                      (fetch 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 STATELASTERROR of STATE with (CDR SESSION))
            (LET [(ERRMSG (CASE (CADR SESSION)
                                ((REJECT)
                                 (LET* ((REASON (CADDR SESSION))
                                         (TYPE (CAR REASON)))
                                   (if (AND (EQ TYPE 'WrongVersionOfService)
                                           (<= (CAADR REASON)
                                                1)
                                           (< (CAADR REASON)
                                                2))
                                     then
                                       ; Server supports old inbasket, but not new
                                       (PRINTOUT PROMPTWINDOW T T "****Note: "
                                         (fetch (MAILSERVER MAILSERVERNAME) of MAILSERVER
                                         )
                                         " does not support new mail protocols." T)
                                       (if (NOT RETURNERRORS)
                                           then (RETURN NIL)))
                                     TYPE))
                                ((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 'NoSuchInbasket)
              (\NSMAIL.FIX.MAILBOX.LOCATIONS))
    then
      ;; 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)))

```

(NEWS.NEXTMESSAGE

```

[LAMBDA (MAILBOX)
  (LET [(NEXT (NEW.INBASKET.CALL MAILBOX 'RETRIEVE.ENVELOPES (fetch NSMAILLASTINDEX of MAILBOX)
                                'NEXT
                                (fetch NSMAILSESSION of MAILBOX))
        ;; NEXT = (envelope status index)
        (DESTRUCTURING-BIND (ENVELOPE STATUS INDEX)
          NEXT
          (if (EQ INDEX 0)
              then
                NIL
              else (replace NSMAILLASTINDEX of MAILBOX with INDEX)
                   (replace NSMAILENVTAIL of MAILBOX with ENVELOPE)
                   ; Success
                   T)])

```

; Edited 13-Dec-89 17:27 by bvm

; (fetch NSMAILLASTINDEX of MAILBOX)

; No more messages

; Success

(NEWS.RETRIEVEMESSAGE

```

[LAMBDA (MAILBOX MSGOUTFILE)
  (DECLARE (SPECVARS *ATTACHMENTS* *DISCARDED-PARTS* *ENVELOPE* *FORMAT-STREAM* *HAVE-ATTACHMENTS*
                    *HEADER-EOF* *MSGSTREAM* *RETRIEVAL-ERROR* *TABLE-OF-CONTENTS*))
  ; Edited 16-Jan-90 15:43 by bvm
  ; For the bulk data fn
  (PROG* ((*RETRIEVAL-ERROR* NIL)
          (INDEX (fetch NSMAILLASTINDEX of MAILBOX))
          (*ENVELOPE* (fetch NSMAILENVTAIL of MAILBOX))
          (*TABLE-OF-CONTENTS* (CADR (ASSOC 'TOC *ENVELOPE*)))
          (*MSGSTREAM* MSGOUTFILE)
          (HERE 0)
          *DISCARDED-PARTS* *HAVE-ATTACHMENTS* *ATTACHMENTS* *FORMAT-STREAM* *HEADER-EOF* PARTS-TO-RETRIEVE
          RESULT REPORT)
    (for PAIR in *TABLE-OF-CONTENTS* bind OTHER do (if (FMEMB (CAR PAIR)
                                                                \NSMAIL.GOOD.BODY.PARTS)
                then
                  ; we read this fine
                elseif (SETQ OTHER (ASSOC (CAR PAIR)
                                           \NSMAIL.DISCARDABLE.BODY.PARTS
                                           ))
                  then (push *DISCARDED-PARTS* OTHER)
                  else ; Will need to arrange for an attachment
                      (SETQ *HAVE-ATTACHMENTS* T)))
    (if (NOT *HAVE-ATTACHMENTS*)
        then
          ; Write directly to MSGOUTFILE. Note where we are in case we
          ; have to retry
          (SETQ HERE (GETFILEPTR *MSGSTREAM*))
        [if *DISCARDED-PARTS*
          then
            ; Ordinarily we retrieve everything (PARTS-TO-RETRIEVE =
            ; NIL), but if there were parts we like to ignore, we can skip
            ; these.
            (SETQ PARTS-TO-RETRIEVE (for PAIR in *TABLE-OF-CONTENTS* as INDEX from 0 collect INDEX
                                      unless (ASSOC (CAR PAIR)
                                                      *DISCARDED-PARTS*])
            RETRY
            [if *HAVE-ATTACHMENTS*
              then (SETQ *MSGSTREAM* (OPENSTREAM '{NODIRCORE} 'BOTH)
              (if (SETQ REPORT (CADR (ASSOC 'REPORT *ENVELOPE*)))
                  then
                    ; This is a delivery report. What a cruffy way to represent it
                    (SETQ *ENVELOPE* (\NSMAIL.HANDLE.DELIVERY.REPORT *MSGSTREAM* REPORT *ENVELOPE*))
                    (if (NULL *TABLE-OF-CONTENTS*)
                        then
                          ; No body, e.g., a bad dl member report
                          (GO FINISH)
                        else
                          ; Some message parts will follow the report
                          (PRINTOUT *MSGSTREAM* T "- - - - -" T)))
                    (if (NEQ (CAAR *TABLE-OF-CONTENTS*)
                            (\NSMAIL.BODY.PART.TYPE HEADING))
                        then (HELP "First body part is not heading" *TABLE-OF-CONTENTS*))
                    (SETQ RESULT (COURIER.CALL (fetch NSMAILSTREAM of MAILBOX)

```

```

'NEW.INBASKET
'RETRIEVE.BODY.PARTS INDEX PARTS-TO-RETRIEVE (FUNCTION \NSMAIL.READ.BODY.PARTS)
(fetch NSMAILSESSION of MAILBOX)
'RETURNERRORS))
(if (EQ (CAR (LISTP RESULT))
'ERROR)
then (\NSMAIL.SIGNAL.ERROR RESULT MAILBOX 'NEW.INBASKET 'RETRIEVE.BODY.PARTS)
(if *HAVE-ATTACHMENTS*
then (SETQ *ATTACHMENTS* NIL)
else (SETFILEPTR MSGOUTFILE HERE))
(SETQ *RETRIEVAL-ERROR* NIL)
(GO RETRY))
(COND
(*RETRIEVAL-ERROR* (printout *MSGSTREAM* T *RETRIEVAL-ERROR* T)))
(if *FORMAT-STREAM*
then ; This is a TEdit formatted message
(LA.ADJUST.FORMATTING *FORMAT-STREAM* *MSGSTREAM* (- *HEADER-EOF* HERE)))
(if *HAVE-ATTACHMENTS*
then (SETQ *MSGSTREAM* (OPENTEXTSTREAM *MSGSTREAM* NIL NIL NIL (LIST 'FONT LAFITEDISPLAYFONT)))
[LET ((ATTACHPOINT (TEDIT.FIND *MSGSTREAM* "
Attachment: " 1)))
(SETQ ATTACHPOINT (if ATTACHPOINT
then ; Insert object at end of this line
(+ ATTACHPOINT 14)
else ; Shouldn't happen
(+ (TEDIT.FIND *MSGSTREAM* "
" 1)
2)))
(for AT in *ATTACHMENTS*
do (LET (TYPE)
(SETFILEPTR AT 4) ; Skip the version number (LONGCARDINAL). Next comes
; SEQUENCE Filing.Attribute
; unknown
[to (\WIN AT) bind X ATTR
do (if (EQ (SETQ ATTR (COURIER.READ AT NIL 'LONGCARDINAL))
(\NSMAIL.ATTRIBUTE.TYPE BodyType))
then (\WIN AT)
[SETQ TYPE (\TYPE.FROM.FILETYPE (COURIER.READ AT NIL
'LONGCARDINAL])
else (COURIER.SKIP.SEQUENCE AT NIL 'UNSPECIFIED)
(TEDIT.INSERT.OBJECT (\MAILOBJ.CREATE AT TYPE (GETFILEPTR AT))
*MSGSTREAM* ATTACHPOINT)
;; Would like the following to be (COERCETEXTOBJ OUTSTREAM 'FILE MSGOUTFILE) but Tedit has a bug
(COPYBYTES (OPENSTREAM (COERCETEXTOBJ *MSGSTREAM* 'FILE)
'INPUT)
MSGOUTFILE))
FINISH
(push (fetch NSMAILENVELOPES of MAILBOX)
INDEX])

```

(\NSMAIL.READ.BODY.PARTS

```

[LAMBDA (BULKSTREAM) ; Edited 14-Aug-90 16:13 by bvm
(DECLARE (SPECVARS *ATTACHMENTS* *DISCARDED-PARTS* *ENVELOPE* *FORMAT-STREAM* *HAVE-ATTACHMENTS*
*HEADER-EOF* *MSGSTREAM* *TABLE-OF-CONTENTS* *BODY-OFFSET*))
;; Bulk data handler for RetrieveBodyParts call. We see the body parts, one directly after the other, per toc.
;; I hope the heading part is first
(for PAIR in *TABLE-OF-CONTENTS* as INDEX from 0 bind (START _ (GETFILEPTR BULKSTREAM))
(*BODY-OFFSET* _ 0)
END HAVETEXT DISCARDING HEADERFIELDS FORWARDINFO FINFO
FORWARDSTREAM PART-TYPE PART-LENGTH
unless (ASSOC (SETQ PART-TYPE (CAR PAIR))
*DISCARDED-PARTS*)
do ;; Assertion: START = (getfileptr bulkstream)
(SETQ PART-LENGTH (CADR PAIR))
[if DISCARDING
then ; We already ate some of this, have to skip the rest
(if (> (SETQ DISCARDING (- DISCARDING PART-LENGTH))
0)
then ; We've eaten the entire part, keep discarding
else ; We've eaten all but -DISCARDING bytes
(SETFILEPTR BULKSTREAM (SETQ START (- START DISCARDING)))
(SETQ DISCARDING NIL))
else (SETQ END (+ START PART-LENGTH))
[SETQ FINFO (find F in FORWARDINFO suchthat ; See if this is a forwarded part
(FMEMB INDEX (fetch (FORWARD PARTS) of F]
[SELECTC PART-TYPE
((\NSMAIL.BODY.PART.TYPE HEADING) ; The heading = Sequence of Heading Attribute
(CL.MULTIPLE-VALUE-SETQ (HEADERFIELDS *FORMAT-STREAM* FORWARDINFO)
(\NSMAIL.READ.HEADING BULKSTREAM END))
(\NSMAIL.NEW.PRINT.HEADING *MSGSTREAM* HEADERFIELDS *ENVELOPE*)
; Print your basic heading. May set *BODY-OFFSET*

```

```

    (if *DISCARDED-PARTS*
      then
        (MAPRINT (CL:REMOVE-DUPLICATES (MAPCAR *DISCARDED-PARTS*
          (FUNCTION CADR))
          :TEST
          'STRING-EQUAL)
          *MSGSTREAM* "Discarded-Parts: " NIL ", ")
        (TERPRI *MSGSTREAM*))
    (if *HAVE-ATTACHMENTS*
      then
        (PRINTOUT *MSGSTREAM* "
          Attachment: " T))
        (TERPRI *MSGSTREAM*)
        (SETQ *HEADER-EOF* (GETFILEPTR *MSGSTREAM*))
        [if FORWARDINFO
          then
            (SETQ FORWARDSTREAM (OPENSTREAM '{NODIRCORE} 'BOTH])
            ((LIST (\NSMAIL.BODY.PART.TYPE MULTINATIONAL.NOTE)
              (\NSMAIL.BODY.PART.TYPE NSTEXTFILE)
              (\NSMAIL.BODY.PART.TYPE IA5.NOTE))
              ; This is text
            (LET ((OUTSTREAM *MSGSTREAM*)
              (OFFSET *BODY-OFFSET*)
              FORWARDSTART)
              (if FINFO
                then
                  (SETQ FORWARDSTART (GETFILEPTR (SETQ OUTSTREAM FORWARDSTREAM)))
                  (SETQ OFFSET 0)
                else
                  (if HAVETEXT
                    then
                      (PRIN3 "
                        - - - - -
                        " *MSGSTREAM*)
                      (SETQ HAVETEXT T))
                    else (SETQ HAVETEXT T)))
              (SELECTC PART-TYPE
                ((\NSMAIL.BODY.PART.TYPE MULTINATIONAL.NOTE)
                  ; Xerox character set--just copy.
                  (SETFILEPTR BULKSTREAM (+ START OFFSET))
                  (COPYBYTES BULKSTREAM OUTSTREAM (- PART-LENGTH OFFSET)))
                ((\NSMAIL.BODY.PART.TYPE IA5.NOTE)
                  ; ia5 takes a little bit of conversion. Note that the skip case never
                  ; happens here
                  (\NSMAIL.COPY.IA5 BULKSTREAM OUTSTREAM PART-LENGTH))
                ((\NSMAIL.BODY.PART.TYPE NSTEXTFILE)
                  ; nstextfile--decode serialized file
                  (\NSMAIL.COPY.NSTEXTFILE BULKSTREAM OUTSTREAM END OFFSET))
                NIL)
              (if FINFO
                then
                  ; Record where the text went
                  (push (fetch (FORWARD MAP) of FINFO)
                    (LIST INDEX FORWARDSTART (- (GETFILEPTR FORWARDSTREAM)
                      FORWARDSTART)))
                else
                  ; We've finished whatever skipping we were going to do.
                  (SETQ *BODY-OFFSET* 0)))
            (LET [(BODY (OPENSTREAM '{NODIRCORE} 'BOTH])
              ; Parts we don't handle become opaque attachments
              (if (OR (EQL PART-TYPE (\NSMAIL.BODY.PART.TYPE VPFOLDER))
                (EQL PART-TYPE (\NSMAIL.BODY.PART.TYPE VPDOCUMENT))
                (EQL PART-TYPE (\NSMAIL.BODY.PART.TYPE OTHERNSFILE)))
                then
                  ; It's already serialized
                  (COPYBYTES BULKSTREAM BODY PART-LENGTH)
                else
                  ; for now, make a serialized file
                  (COURIER.WRITE BODY \SERIALIZED.FILE.VERSION NIL 'LONGCARDINAL)
                  ; version
                  (\WOUT BODY 1)
                  ; Length of attribute sequence
                  (\NSMAIL.WRITE.ATTRIBUTE BODY 'BodyType (if (EQL PART-TYPE (
                    \NSMAIL.BODY.PART.TYPE
                    INTERPRESS))
                      then (CONSTANT (\FILETYPE.FROM.TYPE
                        'INTERPRESS))
                      else (+ PART-TYPE 100000000)))
                  (COURIER.WRITE BODY (COURIER.WRITE.STREAM.UNSPECIFIED BODY BULKSTREAM START END)
                    NIL
                    'BOOLEAN)
                  ; StreamOfUnspecified followed by lastBytelsSignificant
                  (\WOUT BODY 0)
                  ; no children
                  )
              (push *ATTACHMENTS* BODY)
              (if FINFO
                then
                  ; So we can refer to this later as attachment #n
                  (push (fetch (FORWARD MAP) of FINFO)
                    (LIST INDEX (LENGTH *ATTACHMENTS*)
                      (HELP (CL:FORMAT NIL "Body part ~A wrong length: parsed as ~D, should have been ~D"
                        PART-TYPE (+ PART-LENGTH (- START END))

```

```

PART-LENGTH)
(CL:FORMAT NIL "Type 'RETURN' to " (if (> START END)
                                     then "flush rest of message"
                                     else "flush unread portion"))
(if (> START END)
    then (SETQ DISCARDING (- START END))
    else (SETFILEPTR BULKSTREAM (SETQ START END)
finally (if FORWARDINFO
            then ;; At this point we have written all the original parts. Now walk thru the Forwarding info and write those messages
              (LET ((*NSMAIL-DISPLAY-TRANSPORT-ID* NIL)
                    (*NSMAIL-DISPLAY-POSTMARK* NIL)) ; Those fields are boring in forwarded mail
                (\NSMAIL.EMIT.FORWARDING FORWARDINFO FORWARDSTREAM *MSGSTREAM* NIL)))
              ;; Return NIL to let Courier result show thru
              (RETURN NIL])

```

(\NSMAIL.COPY.IA5

[LAMBDA (INSTREAM OUTSTREAM NBYTES)

; Edited 22-Dec-89 18:06 by bvm

;; Convert NBYTES of ia5 text on INSTREAM to Xerox charset on OUTSTREAM

```

(while (>= (SETQ NBYTES (SUB1 NBYTES))
        0)
  bind CH do (SELCHARQ (SETQ CH (\BIN INSTREAM))
                (CR) ; CR followed by some number of lfs indicates line breaks
                [bind GOT1 while (AND (>= (SETQ NBYTES (SUB1 NBYTES))
                                          0)
                                      (EQ (SETQ CH (\BIN INSTREAM))
                                          (CHARCODE LF)))
                do ; One eol for each lf
                  (\BOUT OUTSTREAM (CHARCODE CR))
                  (SETQ GOT1 T)
                finally (if (NOT GOT1)
                            then ; Naked CR? Well, go ahead and print one anyway--we don't
                                ; know how else to do it
                                (\BOUT OUTSTREAM (CHARCODE CR))
                            (if (< NBYTES 0)
                                then ; Text ended in eol
                                    (RETURN)))
                NIL)
  (\BOUT OUTSTREAM CH])

```

(\NSMAIL.COPY.NSTEXTFILE

[LAMBDA (INSTREAM OUTSTREAM END OFFSET)

; Edited 22-May-90 10:37 by bvm

```

;; Copies the serialized text file from INSTREAM to OUTSTREAM. If there's a formatting item, sets *FORMAT-STREAM*. Just in case of trouble,
;; END is the file pointer where we expect the file to end. If OFFSET is specified, it is an initial number of bytes to skip.

```

```

(\NSMAIL.CHECK.SERIALIZED.VERSION INSTREAM) ; Now Sequence of Filing.Attribute
[to (\WIN INSTREAM) bind TYPE do (SETQ TYPE (COURIER.READ INSTREAM NIL 'LONGCARDINAL))
  (if (AND (EQL TYPE (\NSMAIL.ATTRIBUTE.TYPE LispFormatting))
          (NOT *FORMAT-STREAM*))
      then ; Read formatting
        [\NSMAIL.READ.STRING.AS.STREAM INSTREAM (SETQ *FORMAT-STREAM*
                                                         (OPENSTREAM ' {NODIRCORE}
                                                         ' BOTH))
      else ; Skip over the value
        (COURIER.SKIP.SEQUENCE INSTREAM NIL 'UNSPECIFIED])

```

;; Now read the text content. This is adapted from \nsmail.read.serialized.content

```

(if (NOT OFFSET)
    then (SETQ OFFSET 0))
(bind LASTSEGMENT? BYTE BYTECOUNT do (SETQ LASTSEGMENT? (NEQ (\WIN INSTREAM)
                                                                0))
  (COND
    [(NEQ (SETQ BYTECOUNT (UNFOLD (\WIN INSTREAM)
                                    BYTESPERWORD))
          0)
     (if (AND (> OFFSET 0)
              (LET ((SKIPLength (MIN OFFSET BYTECOUNT))
                    ; How much of this segment to skip
                    (SETFILEPTR INSTREAM (+ (GETFILEPTR INSTREAM)
                                              SKIPLength))
                    (SETQ OFFSET (- OFFSET SKIPLength))
                    (EQ (SETQ BYTECOUNT (- BYTECOUNT SKIPLength))
                        0)))
          then ; We skipped the entire segment
            (if LASTSEGMENT?
                then ; Have to consume the lastByteIsSignificant flag
                    (\WIN INSTREAM))
            else (COPYBYTES INSTREAM OUTSTREAM (SUB1 BYTECOUNT))
                  (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
                                (\ABOUT OUTSTREAM BYTE]
                                (LASTSEGMENT? ; Null body. Throw out the lastBytelsSignificant flag
                                (\WIN INSTREAM)))

repeatuntil LASTSEGMENT?)
(LET ((NCHILDREN (\WIN INSTREAM)))
  (if (> NCHILDREN 0)
    then (HELP "nsTextFile has children!! -- return to skip them" NCHILDREN)
    (SETFILEPTR INSTREAM END]))

(\NSMAIL.READ.HEADING
[LAMBDA (BULKSTREAM HEADING-END)
; Edited 21-Dec-89 17:09 by bvm
;; Read a Heading body part, which consists of Sequence of Heading Attribute. Returns 4 values: headerfields (an alist), formatstream (if there
;; was tedit formatting item), forwardInfo (if there was a forwarding structure), malformedP (if we had to advance the file pointer manually to
;; HEADING-END
(LET
  (TYPE VALUE HEADERFIELDS TYPEINFO DISCARDED FORMATSTREAM FORWARDINFO MALFORMED COURIERTYPE)
  [to (\WIN BULKSTREAM)
  do
    (SETQ TYPE (COURIER.READ BULKSTREAM NIL 'LONGCARDINAL))
    (COND
      ((NOT (find old TYPEINFO in \NSMAIL.HEADING.ATTRIBUTES suchthat (EQ (CADR TYPEINFO)
                                     TYPE))))
      ; We don't understand this attribute

      (if NSMAILDEBUGFLG
        then (push DISCARDED TYPE))
      (COURIER.SKIP.SEQUENCE BULKSTREAM NIL 'UNSPECIFIED))
    ((EQ (SETQ TYPE (CAR TYPEINFO))
      'LispFormatting)
      ; Save the formatting so we can munge it
      (SETQ FORMATSTREAM (OPENSTREAM '{NODIRCORE} 'BOTH))
      (\NSMAIL.READ.STRING.AS.STREAM BULKSTREAM FORMATSTREAM))
    (T
      (LET
        ((VALUE-END (+ (UNFOLD (\WIN BULKSTREAM)
                               BYTESPERWORD)
          (GETFILEPTR BULKSTREAM))))
        (COURIERTYPE (CADDR TYPEINFO))
        HERE)
        ; Note careful order of args to +
        (if (EQ TYPE 'ForwardedHeadings)
          then (SETQ FORWARDINFO (\NSMAIL.READ.FORWARDING BULKSTREAM VALUE-END))
          else (CL:MULTIPLE-VALUE-BIND (VALUE CONDITION)
            (IGNORE-ERRORS (COURIER.READ BULKSTREAM 'NEW.MAILTRANSPORT COURIERTYPE))
            [if (OR CONDITION (NOT (EQL (SETQ HERE (GETFILEPTR BULKSTREAM))
                                         VALUE-END)))]
            then (if (NOT CONDITION)
              then (SETQ CONDITION "wrong length"))
              (if NSMAILDEBUGFLG
                then (HELP (CL:FORMAT NIL "Error reading attribute ~A: ~A" TYPE CONDITION)))
                (push HEADERFIELDS (CONS (MKSTRING TYPE)
                  (CL:FORMAT NIL "XNS encoding error: ~A" CONDITION)))
                (if (< HERE VALUE-END)
                  then (SETFILEPTR BULKSTREAM VALUE-END)
                  elseif (AND (> HERE VALUE-END)
                    (< HERE HEADING-END))
                  then (SETFILEPTR BULKSTREAM HEADING-END)
                    (push HEADERFIELDS '("Header-Errors"
                      . "Malformed XNS heading, some fields may be missing."))
                      ; Exit this heading reader loop
                      (RETURN (SETQ MALFORMED T)))
                    ; Save field
                    (push HEADERFIELDS (CONS TYPE (if (EQ TYPE 'Immutable)
                      then
                        ; Strange null-valued type
                        "True"
                      elseif (LISTP COURIERTYPE)
                        then (if (EQUAL COURIERTYPE '(SEQUENCE IP.MESSAGEID)
                          )
                          then (MAPCAR VALUE (FUNCTION
                            \NSMAIL.TRANSLATE.IP.MESSAGEID
                            ))
                          else VALUE)
                        else (SELECTQ COURIERTYPE
                          (TIME (\NSMAIL.GDATE VALUE))
                          (IP.MESSAGEID (\NSMAIL.TRANSLATE.IP.MESSAGEID
                            VALUE))
                          VALUE)))
                    VALUE]]]
            [if DISCARDED
              then (push HEADERFIELDS (CONS "Discarded-Fields" (CONCATLIST (CDR (for D in (REVERSE DISCARDED)
                join (LIST ", " D]
                (CL:VALUES HEADERFIELDS FORMATSTREAM FORWARDINFO MALFORMED]))

```

(\NSMAIL.PARSE.ANNOTATION

[LAMBDA (ANNOTATION OUTSTREAM HEADERFIELDS)

; Edited 21-Dec-89 13:10 by bvm

;; ANNOTATION is the value of the TextAnnotation heading. We parse it and print it to OUTSTREAM. HEADERFIELDS is an alist of other
 ;; headers the caller will be printing.

(bind (LEN _ (NCHARS ANNOTATION))

(START _ 1)

(NEXT _ 1)

CR while (SETQ CR (STRPOS " " ANNOTATION NEXT))

do (CASE (AND (< CR LEN) (CL:CHAR ANNOTATION CR))

((#\Space #\Tab)

; Whitespace denoting continuation line

)

(T (\NSMAIL.EMIT.ANNOTATION (SUBSTRING ANNOTATION START (SUB1 CR))

OUTSTREAM HEADERFIELDS)

(SETQ START (ADD1 CR)))

(SETQ NEXT (ADD1 CR))

finally (\NSMAIL.EMIT.ANNOTATION (SUBSTRING ANNOTATION START) OUTSTREAM HEADERFIELDS])

(\NSMAIL.EMIT.ANNOTATION

[LAMBDA (STR OUTSTREAM HEADERFIELDS)

; Edited 10-Jul-90 15:55 by bvm

(DECLARE (SPECVARS *ORIGINAL-DATE*))

;; Print extra field STR to OUTSTREAM. We don't know exactly what it looks like, so we need to ensure that it is syntactically ok. If it is one of
 ;; HEADERFIELDS, we make sure to rename it to avoid a clash. If it is the Date field, we print it and set *ORIGINAL-DATE* to the value portion.

(PROG (I LEN FIELD)

(if (AND STR (> (SETQ LEN (NCHARS STR)) 0))

then [if [NOT (SETQ FIELD (for old I from 0 to (SUB1 LEN)

do (CASE (CL:CHAR STR I)

(#\: ; valid field

(RETURN (SUBSTRING STR 1 I)))

((#\Space #\Tab)

; Space before colon? Malformed

(RETURN NIL))]

; Malformed field

then

(PRIN3 "Other-Field: " OUTSTREAM)

elseif (CL:ASSOC FIELD HEADERFIELDS :TEST 'STRING-EQUAL)

then

; We already have a field of this name, so rename it

(if (AND (< I (- LEN 2))

(EQL (CL:CHAR STR (ADD1 I))

#\Tab))

then

; field:<tab> looks a little weird when we add text to the front

(CL:SETF (CL:CHAR STR (ADD1 I))

#\Space))

(PRIN3 "Original-" OUTSTREAM)

elseif (STRING-EQUAL FIELD "Date")

then (SETQ *ORIGINAL-DATE* (LA.TRIM.WHITESPACE (SUBSTRING STR (+ I 2]

(PRIN3 STR OUTSTREAM)

(TERPRI OUTSTREAM]))

(LA.TRIM.WHITESPACE

[LAMBDA (STR)

; Edited 14-May-90 16:35 by bvm

(CL:STRING-TRIM '(\Space #\Tab) STR])

(\NSMAIL.READ.FORWARDING

[LAMBDA (INSTREAM VALUE-END)

; Edited 21-Dec-89 18:39 by bvm

;; Read the attribute ForwardedHeadings = Sequence of ForwardedMessageInfo. We do this instead of a straight COURIER.READ so that we can
 ;; play with the headings field. Returns NIL if the attribute is malformed.

(to (\WIN INSTREAM) collect (create FORWARD

ENVELOPE _ (COURIER.READ INSTREAM 'NEW.MAILTRANSPORT 'ENVELOPE)

HEADINGS _ (CL:MULTIPLE-VALUE-BIND (HEADINGS FORMATSTREAM FORWARDINFO

MALFORMED)

(\NSMAIL.READ.HEADING INSTREAM VALUE-END)

(if MALFORMED

then (RETURN NIL)

else

;; Note that we ignore FORWARDINFO (not allowed anyway, as messages are not quite recursive)
 ;; and FORMATSTREAM (who would have had it anyway, though it would be cute to be able to use
 ;; it).

HEADINGS))

PARTS _ (COURIER.READ.SEQUENCE INSTREAM 'NEW.MAILTRANSPORT

'BODY.PART.INDEX)

PARENT _ (if (NEQ (\WIN INSTREAM)

0)

then ; Open coding of (choice (null 0 (record)) (nested 1 cardinal))

(\WIN INSTREAM])

(\NSMAIL.NEW.PRINT.HEADING

```

[LAMBDA (OUTSTREAM HEADERFIELDS ENVELOPE) ; Edited 26-Sep-90 11:35 by bvm
;; Compose message header from HEADERFIELDS and ENVELOPE, printing to OUTSTREAM.
(PROG (*ORIGINAL-DATE* ORIGIDATE POSTED.DATE ORIGINATOR RETURN-TO VALUE TYPE SENDER FROMFIELD FULLFROMFIELD)
(DECLARE (SPECVARS *ORIGINAL-DATE* *BODY-OFFSET*))
[for PAIR in (SETQ HEADERFIELDS (REVERSE HEADERFIELDS))
do
(CASE (CAR PAIR)
(Sender (SETQ SENDER (CDR PAIR)))
(From
(SETQ FULLFROMFIELD (CDR PAIR))
[COND
(NULL (CDDR PAIR)) ; Only interesting to eliminate if there's only one
(SETQ FROMFIELD (CADR PAIR))
((TextAnnotation newTextAnnotation)
(\NSMAIL.PARSE.ANNOTATION (CDR PAIR)
OUTSTREAM HEADERFIELDS)
(RPLACD PAIR NIL))
(BodyOffset
(SETQ *BODY-OFFSET* (CDR PAIR)) ; Says how much of body duplicates the textannotation
(RPLACD PAIR NIL)))]
;; Look at the envelope to see if there is any additional info we should supply that wasn't in the headers
[for PAIR in ENVELOPE do (SETQ VALUE (CADR PAIR))
(CASE (SETQ TYPE (CAR PAIR))
(Originator (if (OR (NULL (OR SENDER FROMFIELD))
(NOT (EQUAL.RNAMES VALUE (OR SENDER FROMFIELD))
then (SETQ ORIGINATOR VALUE)))
(RETURN.TO.NAME (SETQ RETURN-TO VALUE))
(Message-ID (if *NSMAIL-DISPLAY-TRANSPORT-ID*
then (CL:FORMAT OUTSTREAM "XNS-Transport-ID:
~{~4,'0x~}~%" VALUE)))
(Postmark
(SETQ POSTED.DATE (COURIER.FETCH (NEW.MAILTRANSPORT . POSTMARK)
TIME of VALUE))
[if *NSMAIL-DISPLAY-POSTMARK*
then (CL:FORMAT OUTSTREAM "Postmark: ~A at ~A~%"
(RNAME.TO.STRING (COURIER.FETCH (NEW.MAILTRANSPORT
. POSTMARK)
POSTED.AT of VALUE)
T)
(GDATE POSTED.DATE (DATEFORMAT TIME.ZONE))]]
(if POSTED.DATE
then ; Date is found only in the envelope
(if (AND *ORIGINAL-DATE* (SETQ ORIGIDATE (IDATE *ORIGINAL-DATE*))
(< (IABS (- POSTED.DATE ORIGIDATE))
(TIMES 5 60)))
then ; Text-annotation portion gave a date that is within 5 minutes, so
; don't bother mentioning the posting date.
else (if *ORIGINAL-DATE*
then ; Already have a Date field printed, so this one we'll call
; "Posted-Date"
(PRINTOUT OUTSTREAM "Posted-"))
(PRINTOUT OUTSTREAM "Date: " (\NSMAIL.GDATE POSTED.DATE
T)))
(if (NULL FULLFROMFIELD)
then ; Derive From field from somewhere else
[if SENDER
then (RPLNODE (ASSOC 'Sender HEADERFIELDS)
'From
(LIST SENDER))
(if ORIGINATOR
then (push HEADERFIELDS (CONS 'Sender ORIGINATOR)))
elseif ORIGINATOR
then ; Neither From nor Sender in heading, take it out of envelope
(push HEADERFIELDS (LIST 'From (SETQ SENDER ORIGINATOR))
elseif (NULL SENDER)
then ; From but no Sender
(if ORIGINATOR
then ; ORIGINATOR only set when it's different from From
(push HEADERFIELDS (CONS 'Sender (SETQ SENDER ORIGINATOR)))
else (SETQ SENDER FROMFIELD))
elseif (AND FROMFIELD (EQUAL.RNAMES SENDER FROMFIELD))
then ; Sender is redundant with From--get rid of it, unless the
; envelope originator is different
(RPLACD (ASSOC 'Sender HEADERFIELDS)
ORIGINATOR)
elseif ORIGINATOR
then ; Three distinct fields
(push HEADERFIELDS (CONS 'Originator ORIGINATOR)))
(if (AND RETURN-TO (OR (NULL SENDER)
(NOT (EQUAL.RNAMES RETURN-TO SENDER)))
*NSMAIL-DISPLAY-ERRORS-TO*)
then ; Usually same as originator, so we omit. (NULL SENDER) is only true when there's no originator in envelope, allegedly illegal

```

```

(push HEADERFIELDS (CONS 'Errors-To RETURN-TO)))
(if (NOT (type? NSNAME SENDER))
    then
        (SETQ SENDER NIL)) ; Can't resolve domain/orgs against this
(for PAIR in [SORT HEADERFIELDS (FUNCTION (LAMBDA (X Y)
;; X sorts before Y if X is in the well-known order and either Y appears after it or doesn't appear at
;; all. Non-symbols sort after everything
(AND (LITATOM (CAR X))
(OR (NOT (LITATOM (CAR Y)))
(AND (SETQ X (FMEMB (CAR X)
NSMAIL.HEADER.ORDER))
(OR (FMEMB (CAR Y)
X)
(NULL (FMEMB (CAR Y)
NSMAIL.HEADER.ORDER]
when (SETQ VALUE (CDR PAIR))
do
(printout OUTSTREAM (SETQ TYPE (CAR PAIR))
": ")
(CASE TYPE
((From To cc bcc Reply-to)
(\NSMAIL.NEW.PRINT.NAMES
VALUE OUTSTREAM
(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
NIL) ; always full-qualified
SENDER)))
((Sender Originator Errors-To) (printout OUTSTREAM (RNAME.TO.STRING VALUE T)))
(T [if (LISTP VALUE)
then
; List of things we'll print as each thing separated by spaces
; (e.g., References)
(SETQ VALUE (CONCATLIST (CDR (for X in VALUE join (LIST " " X)
(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)))
[bind (CR _ 1) while (SETQ CR (STRPOS "
" VALUE CR))
do
; Given internal CR, have to make sure subsequent lines are
; continuation lines, i.e., start with whitespace.
(SELCHARQ (NTHCHARCODE VALUE (ADD1 CR))
((SPACE TAB)
(SETQ CR (ADD1 CR)))
(PROGN
; Not followed by whitespace, so print this much (including cr),
; then a tab.
(PRIN3 (SUBSTRING VALUE 1 CR)
OUTSTREAM)
(PRINTCCODE (CHARCODE TAB)
OUTSTREAM)
(SETQ VALUE (SUBSTRING VALUE (ADD1 CR)))
(SETQ CR 1)
(PRIN3 VALUE OUTSTREAM)))
(TERPRI OUTSTREAM])

```

(\NSMAIL.NEW.PRINT.NAMES

```

[LAMBDA (RNames OUTSTREAM DEFAULTNAME)
(for NAME in RNames bind (FIRSTTIME _ T)
ORGDIFFERS

```

; Edited 4-Apr-90 17:32 by bvm

```

do (if (type? NSNAME NAME)
then (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)
    else (PRIN3 (RNAME.TO.STRING NAME)
                OUTSTREAM])

```

(\NSMAIL.EMIT.FORWARDING

[LAMBDA (FORWARDINFO FORWARDSTREAM OUTSTREAM PARENT-INDEX) ; Edited 22-May-90 10:41 by bvm

;; Recursively emit Forwarded body structure. In this pass, we print all the body parts subsidiary to the PARENT-INDEXth item, or the top level
 ;; items if PARENT-INDEX is nil.

```

(for FINFO in FORWARDINFO as I from 0 bind NTHTIME when (EQ (fetch (FORWARD PARENT) of FINFO)
                                                             PARENT-INDEX)
do ;; This bit of forwarding info describes a child of PARENT-INDEX
  [LET ((*BODY-OFFSET* 0))
    (DECLARE (SPECVARS *BODY-OFFSET*)) ; set by \nsmail.new.print.heading
    (TERPRI OUTSTREAM)
    (PRIN3 (if NTHTIME
              then ; "Next Message"
                  (CADDR LAFITEFORWARDSTRINGS)
              else (SETQ NTHTIME T) ; "Begin Forwarded Messages"
                  (CADDR LAFITEFORWARDSTRINGS))
            OUTSTREAM)
    (TERPRI OUTSTREAM)
    (\NSMAIL.NEW.PRINT.HEADING OUTSTREAM (fetch (FORWARD HEADINGS) of FINFO)
    (fetch (FORWARD ENVELOPE) of FINFO)) ; Print header of this part
    (TERPRI OUTSTREAM)
    (for INDEX in (fetch (FORWARD PARTS) of FINFO) bind (MAP _ (fetch (FORWARD MAP) of FINFO))
                  MAPENTRY NTHPART
    do (if NTHPART
        then ; Yet another body part
            (PRIN3 "
            - - - - -
            " OUTSTREAM)
        else (SETQ NTHPART T))
      (if [NOT (SETQ MAPENTRY (CDR (ASSOC INDEX MAP)
            then (PRIN3 "[Missing part]
            " OUTSTREAM)
        elseif (CDR MAPENTRY)
            then ; (start length)
                (SETFILEPTR FORWARDSTREAM (+ (CAR MAPENTRY)
                                                *BODY-OFFSET*))
                (COPYBYTES FORWARDSTREAM OUTSTREAM (CADDR MAPENTRY))
                (SETQ *BODY-OFFSET* 0)
            else ; (attachment#)
                (if (CL:FORMAT OUTSTREAM "[See Attachment #~D]~%" (CAR MAPENTRY)
                    ; If there are children, do them
                    (\NSMAIL.EMIT.FORWARDING FORWARDINFO FORWARDSTREAM OUTSTREAM I)
                finally (if NTHTIME
                        then ; Yes, we printed some parts, so time for "End Forwarded
                        ; Messages"
                            (TERPRI OUTSTREAM)
                            (PRIN3 (CADDR LAFITEFORWARDSTRINGS)
                                OUTSTREAM])

```

(\NSMAIL.GDATE

[LAMBDA (TIME) ; Edited 11-Jul-90 18:03 by bvm
(GDATE TIME (DATEFORMAT SPACES TIME.ZONE))

(\NSMAIL.TRANSLATE.IP.MESSAGEID

[LAMBDA (ID) ; Edited 11-May-90 10:45 by bvm

```

(LET ((RNAME (COURIER.FETCH (NEW.MAILTRANSPORT . IP.MESSAGEID)
                            ORIGINATOR of ID))
      (USTRING (COURIER.FETCH (NEW.MAILTRANSPORT . IP.MESSAGEID)
                              UNIQUESTRING of ID)))
  (if (NOT (NULL.NSNAME RNAME))
      then ; Really has name
          (CONCAT #\< USTRING #\* (RNAME.TO.STRING RNAME T)
                  #\>)
      elseif (AND (EQ (CL:CHAR USTRING 0)
                     #\<)
               (EQ (CL:CHAR USTRING (SUB1 (NCHARS USTRING)))

```

then ; Bizarre

```

    (if BADNAMES
      then (PRINTOUT OUTSTREAM "Undeliverable mail" T T)
            (CL:FORMAT OUTSTREAM "This message could not be delivered to the following
              recipient~P:" (LENGTH BADNAMES))
            ; Strictly a delivery report
      else (PRINTOUT OUTSTREAM "Delivery report"))
    (T ; Shouldn't happen
      (PRINTOUT OUTSTREAM "Erroneous (non-)delivery report" T T REPORT)))
    (PRINTOUT OUTSTREAM T T)
    (for PAIR in BADNAMES do (PRINTCCODE (CHARCODE TAB)
      OUTSTREAM)
      (PRINTOUT OUTSTREAM (\NSMAIL.RECIPIENT.NAME (COURIER.FETCH (NEW.MAILTRANSPORT
        . NON.DELIVERED.RECIPIENT)
        RECIPIENT of PAIR))
        " -- "
        (COURIER.FETCH (NEW.MAILTRANSPORT . NON.DELIVERED.RECIPIENT)
        REASON of PAIR)
        T))

    (if GOODNAMES
      then ; A delivery report
        (if BADNAMES
          then (TERPRI OUTSTREAM))
        (CL:FORMAT OUTSTREAM "This message was delivered to the following recipient~P:~2%"
          (LENGTH GOODNAMES))
        (for PAIR in GOODNAMES do (PRINTCCODE (CHARCODE TAB)
          OUTSTREAM)
          (PRINTOUT OUTSTREAM (\NSMAIL.RECIPIENT.NAME (COURIER.FETCH
            (NEW.MAILTRANSPORT
              . DELIVERED.RECIPIENT)
            RECIPIENT of PAIR))
            " at "
            (\NSMAIL.GDATE (COURIER.FETCH (NEW.MAILTRANSPORT
              . DELIVERED.RECIPIENT)
              WHEN of PAIR)
              (DATEFORMAT TIME.ZONE))
              T)))

    OLD.ENVELOPE])

```

(\NSMAIL.RECIPIENT.NAME

[LAMBDA (RECIPIENT)

; Edited 4-Apr-90 17:26 by bvm

;; Printable rep for a MailTransport.Recipient

```

(RNAME.TO.STRING (COURIER.FETCH (NEW.MAILTRANSPORT . RECIPIENT)
  NAME of RECIPIENT)
  T))

```

(NEW.INBASKET.CALL

[CL:LAMBDA (MAILBOX PROCEDURE &REST ARGS)

; Edited 13-Dec-89 17:17 by bvm

```

(PROG ((STREAM (fetch NSMAILSTREAM of MAILBOX))
  RESULT)
  LP (if (AND (EQ [CAR (LISTP (SETQ RESULT (CL:APPLY (FUNCTION COURIER.CALL)
    STREAM
    'NEW.INBASKET PROCEDURE ARGS]
    'ERROR)
    (CASE (CAR (LAST ARGS))
      (NOERROR NIL)
      (RETURNERRORS
        (EQ (CADR RESULT)
          'STREAM.LOST))
      (T
        T)))
    then (SETQ STREAM (\NSMAIL.SIGNAL.ERROR RESULT MAILBOX 'NEW.INBASKET PROCEDURE))
    (GO LP)
    else (RETURN RESULT])

```

; We'll only handle stream lost--caller gets the rest

; Probably an error was already signaled

(NEWS.CLOSEMAILBOX

[LAMBDA (MAILBOX FLUSH?)

; Edited 18-Dec-89 17:35 by bvm

[COND

(FLUSH?

; Delete everything we retrieved

(LET ((INDICES (REVERSE (fetch NSMAILENVELOPES of MAILBOX)))

(SESSION (fetch NSMAILSESSION of MAILBOX)))

(while INDICES do

; Delete a message or more. To keep the calls down, try to
; delete consecutive ranges when possible.

(LET* ((START (CAR INDICES))

(END START))

(while (AND (SETQ INDICES (CDR INDICES))

(EQL (CAR INDICES)

(ADD1 END)))

do (SETQ END (ADD1 END)))

(\NEW.INBASKET.CALL MAILBOX 'DELETE

(COURIER.CREATE (NEW.INBASKET . RANGE)

LOW _ START HIGH _ END)

SESSION])

(\NSMAIL.NEW.LOGOFF (fetch NSMAILSTATE of MAILBOX)

(fetch NSMAILSTREAM of MAILBOX))

(\NSMAIL.NEW.LOGOFF

[LAMBDA (STATE STREAM)

; Edited 19-Dec-89 11:08 by bvm

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

(LET [(RESULT (COURIER.CALL STREAM 'NEW.INBASKET 'LOGOFF (fetch STATESESSION of STATE)
'RETURNERRORS])(PROG1 (NEQ (CAR (LISTP RESULT))
'ERROR)(replace STATESESSION of STATE with NIL)
(CLOSEF STREAM))])

)

(RPAQQ \NSMAIL.GOOD.BODY.PARTS (0 5 6 2))

(RPAQQ \NSMAIL.DISCARDABLE.BODY.PARTS ((201 "Tioga formatting")
(202 "Tioga header")))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \NSMAIL.GOOD.BODY.PARTS \NSMAIL.DISCARDABLE.BODY.PARTS)

)

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

(FILESLOAD LAFITE-NSMAIL)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)

)

(DECLARE%: EVAL@COMPILE

(RECORD FORWARD (ENVELOPE HEADINGS PARTS PARENT . MAP))

)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \NSMAIL.BODY.PART.TYPE MACRO [ARGS (COND
((CADR (ASSOC (CAR ARGS)
\NSMAIL.BODY.PART.TYPES)))
(T (ERROR "Unknown body part type" (CAR ARGS))
'IGNOREMACRO])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \NSMAIL.BODY.PART.TYPES \NSMAIL.HEADING.ATTRIBUTES)

)

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

(FILESLOAD (SOURCE)
LAFITE-DECLS)(FILESLOAD (LOADCOMP)
LAFITE-NSMAIL ATBL)(RPAQQ \NSMAIL.CONTENTS.TYPES ((\CT.NULL 0)
(\CT.STANDARD.MESSAGE 4)
(\CT.REPORT 6)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \CT.NULL 0)

(RPAQQ \CT.STANDARD.MESSAGE 4)

(RPAQQ \CT.REPORT 6)

(CONSTANTS (\CT.NULL 0)
(\CT.STANDARD.MESSAGE 4)
(\CT.REPORT 6))

)

(RPAQQ \NSMAIL.BODY.PART.TYPES

((HEADING 0)
(VPFOLDER 1)

```
(NSTEXTFILE 2)
(VPDOCUMENT 3)
(OTHERNSFILE 4)
(MULTINATIONAL.NOTE 5)
(IA5.NOTE 6)
(PILOTFILE 7)
(G3FAX 8)
(TELETEX 9)
(TELEX 10)
(ISO6937.NOTE 11)
(INTERPRESS 12))
)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA NEW.INBASKET.CALL)
)
```

FUNCTION INDEX

EQUAL.RNAMES	7	\NSMAIL.CHECK.ABORT	13	\NSMAIL.NEW.PRINT.HEADING	23
LA.TRIM.WHITESPACE	22	\NSMAIL.CHECK.ENUMERATION	8	\NSMAIL.NEW.PRINT.NAMES	24
NEW.INBASKET.CALL	27	\NSMAIL.COPY.IA5	20	\NSMAIL.NEW.SEND	9
NEWS.CLOSEMAILBOX	27	\NSMAIL.COPY.NSTEXTFILE	20	\NSMAIL.NEW.SEND.PARSE	7
NEWS.NEXTMESSAGE	17	\NSMAIL.EMIT.ANNOTATION	22	\NSMAIL.PARSE.ANNOTATION	22
NEWS.OPENMAILBOX	15	\NSMAIL.EMIT.FORWARDING	25	\NSMAIL.POST.BODY.PART	12
NEWS.POLLNEWMAIL	15	\NSMAIL.GDATE	25	\NSMAIL.READ.BODY.PARTS	18
NEWS.RETRIEVEMESSAGE	17	\NSMAIL.HANDLE.DELIVERY.REPORT ..	26	\NSMAIL.READ.FORWARDING	22
NULL.NSNAME	26	\NSMAIL.MAYBE.QUOTE	26	\NSMAIL.READ.HEADING	21
RNAME.TO.STRING	7	\NSMAIL.NEW.AUTHENTICATE	14	\NSMAIL.READ.RNAME	6
X400.NAME.TO.STRING	7	\NSMAIL.NEW.CHECK	15	\NSMAIL.RECIPIENT.NAME	27
\NS.NEW.READ.ENVELOPE.ITEM	4	\NSMAIL.NEW.CHECKSERVER	13	\NSMAIL.RNAME.LENGTH	6
\NS.NEW.WRITE.ENVELOPE.ITEM	5	\NSMAIL.NEW.FINDSERVER	13	\NSMAIL.TRANSLATE.IP.MESSAGEID ..	25
\NS.READ.HEADING.ATTRIBUTE	5	\NSMAIL.NEW.INVALID.RECIPIENTS ..	11	\NSMAIL.WRITE.RNAME	6
\NS.WRITE.HEADING.ATTRIBUTE	5	\NSMAIL.NEW.LOGOFF	28		
\NSMAIL.BUILD.HEADING	11	\NSMAIL.NEW.PREPARE.ATTACHMENT ..	12		

VARIABLE INDEX

NEWSMAIL-POST-AS-TEXTFILE	14	*NSMAIL-MIN-WILLINGNESS*	14	\NSMAIL.CONTENTS.TYPES	28
NEWSMAIL-REPORT-TYPE	14	*NSMAIL-OP-VECTOR*	14	\NSMAIL.DISCARDABLE.BODY.PARTS ..	28
NSMAIL-ALLOW-DL-RECIPIENTS	14	*NSMAIL-RETURN-CONTENTS*	14	\NSMAIL.GOOD.BODY.PARTS	28
NSMAIL-CACHE-TIMEOUT	14	*NSMAIL-TRACE-SERVERS*	14	\NSMAIL.HEADING.ATTRIBUTES	6
NSMAIL-DISPLAY-ERRORS-TO	14	LAFITEMODELST	28	\NSMAIL.MAX.VP.TYPE	14
NSMAIL-DISPLAY-POSTMARK	14	NSMAIL.PARSEFIELDS	14	\NSMAIL.MIN.VP.TYPE	14
NSMAIL-DISPLAY-TRANSPORT-ID ...	14	\LAPARSE.NSMAIL	14	\NSMAIL.NEW.ENVELOPE.ITEM.TYPES ..	5
NSMAIL-GENERATE-MESSAGE-ID	14	\NSMAIL.BODY.PART.TYPES	28	\NSMAIL.NEW.SERVER.CACHE	14

CONSTANT INDEX

\CT.NULL	28	\CT.REPORT	28	\CT.STANDARD.MESSAGE	28
----------------	----	------------------	----	----------------------------	----

PROPERTY INDEX

HEADING.ATTRIBUTE	6	NEW.ENVELOPE.ITEM	5	NEW.RNAME	7
-------------------------	---	-------------------------	---	-----------------	---

COURIERPROGRAM INDEX

NEW.INBASKET	3	NEW.MAILTRANSPORT	2
--------------------	---	-------------------------	---

MACRO INDEX

\NSMAIL.BODY.PART.TYPE	28
------------------------------	----

RECORD INDEX

FORWARD	28
---------------	----
