

File created: 15-Jun-90 18:05:32 {DSK}<usr>local>lde>lispcore>internal>library>MAILCLIENT.;2

changes to: (VARS MAILCLIENTCOMS)  
(FNS \GV.SENDMESSAGE)

previous date: 9-Nov-89 15:31:24 {DSK}<usr>local>lde>lispcore>internal>library>MAILCLIENT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **MAILCLIENTCOMS**

```
((COMS
  (FNS GV.STARTSEND GV.ADDRECIPIENT GV.CHECKVALIDITY GV.STARTITEM GV.ADDTOITEM GV.SEND MS.EXPAND)
    ; Sending mail
    ; Internal Sending
  (FNS MS.SENDOPERATION \FINDMAILSERVER \MAILSERVERSOCKETS \RECEIVEACK \RESPTOCHECKVAL \RESPTOEXPAND
    \RESPTOSTARTSEND)
  (INITVARS (\MAILIOTIMEOUT NIL)
    (\MAILSERVERENQUIRYSOC 46)
    (\MAILSERVERNAME ' (Maildrop . ms))
    (\MAILSERVERPOLLINGSOC 44)
    (\MAILSERVERSOCKETCACHE)
    (\MAILSERVERRETRIEVALSOC 47))
  (GLOBALVARS \MAILIOTIMEOUT \MAILSERVERENQUIRYSOC \MAILSERVERNAME \MAILSERVERPOLLINGSOC
    \MAILSERVERSOCKETCACHE \MAILSERVERRETRIEVALSOC)
  (ADDVARS (\SYSTEMCACHEVARS \MAILSERVERSOCKETCACHE)))
  (COMS
    ; Receiving mail
  (FNS GV.PORTFROMNAME GV.POLLNEWMAIL GV.OPENMAILBOX GV.NEXTMESSAGE GV.RETRIEVEMESSAGE
    GV.CLOSEMAILBOX)
  (ADDVARS (MAILSERVERTYPES (GV GV.POLLNEWMAIL GV.OPENMAILBOX GV.NEXTMESSAGE GV.RETRIEVEMESSAGE
    GV.CLOSEMAILBOX GV.PORTFROMNAME)))
  (COMS
    ; Not currently used
    (FNS GV.READTOC GV.WRITETOC GV.DELETEMESSAGE))
    ; Internal Receiving
  (FNS MS.RETRIEVEOPERATION \CONNECTTOMAILSERVER \RESPTOOPENMAILBOX \RESPTONEXTMESSAGE
    \RESPTORETRIEVEMESSAGE \RECEIVELONGWORD \CACHED.HOST.NAME)
  (INITVARS (GV.MAILBOX.TIMEOUT 12000)
    (*GV-SHOW-POSTMARK*)
    (\CACHED.HOST.NAMES))
  (ADDVARS (\SYSTEMCACHEVARS \CACHED.HOST.NAMES))
  (GLOBALVARS GV.MAILBOX.TIMEOUT \CACHED.HOST.NAMES))
  (COMS
    ; LAFITEMODE GV
  (ALISTS (LAFITEMODELST GV GRAPEVINE))
  (FNS GV.INIT.MAIL.USER GETMAILSERVEROPS \GV.MAILSERVERTYPE)
  (FNS \GV.SENDMESSAGE \GV.SENDRECIPIENTS)
  (FNS \GV.SEND.PARSE \GV.PARSERECIPIENTS \GV.PARSERECIPIENTS1 \GV.FIND.NON.SPACE
    \GV.PARSE.SIMPLE.RECIPIENT \GV.EXTRACT.FIELD \GV.HANDLE.DL \GV.PARSE.FAILED
    \LAFITE.CHOOSE.REPLYTO)
  (FNS \GV.MESSAGE.P \GV.MESSAGE.FROM.SELF.P GV.MAKEANSWERFORM \GV.DIFFERENCE)
  (ADDVARS (MAILSERVERTYPES)
    (LAFITEDLDIRECTORIES)
    (LAFITE.PERSONAL.VARS LAFITE.GV.FROM.FIELD))
  (INITVARS (ARPANETGATEWAY.REGISTRY 'AG)
    (LAFITEREPLYTOMENU NIL)
    (LAFITEDL.EXT "DL")
    (LAFITE.GV.FROM.FIELD))
  (ADDVARS (LAFITESUPPORT (GV "LafiteSupport.pa"))
    (LAFITEMENUVARS LAFITEREPLYTOMENU))
  (VARS LAFITEREPLYTOMENUITEMS)
  (PROP FILEDEF MAINTAIN))
  (DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS MAILPORT GVMAILPARSE)
    (COMS * MAILCLIENTCONSTANTCOMS)
    [P (CL:PROCLAIM ' (GLOBAL MAILSERVERTYPES ARPANETGATEWAY.REGISTRY LAFITEREPLYTOMENUITEMS
      LAFITEREPLYTOMENU))
      (CL:PROCLAIM ' (CL:SPECIAL *MSGOUTSTREAM*)]
    (FILES (SOURCE)
      LAFITEDECLS)
    (FILES (LOADCOMP)
      GRAPEVINE PUP BSP)
    (LOCALVARS . T))
  (DECLARE%: DONTVAL@LOAD DOCOPY [P (CL:PROCLAIM ' (GLOBAL LAFITE.GV.FROM.FIELD LAFITEDL.EXT
    LAFITEDLDIRECTORIES))
    (CL:PROCLAIM ' (CL:SPECIAL *GV-SHOW-POSTMARK*)]
    (FILES GRAPEVINE)))))
```

;; Sending mail

(DEFINEQ

(**GV.STARTSEND**

[LAMBDA (SENDER KEY RETURN VALIDATEFLG)

(\* bvm%: " 5-Nov-84 15:39")

;;; returns either a socket to use to send the rest of the message on or NIL

```
(LET (SENDINGSOCKET STARTSENDRESULT)
  (COND
    ((NOT (SETQ SENDINGSOCKET (\FINDMAILSERVER))) ; Can't find a maildrop at all
      NIL)
    ((SETQ STARTSENDRESULT (MS.SENDOPERATION \OP.STARTSEND SENDINGSOCKET [LIST (\CHECKNAME SENDER)
                                          (\CHECKKEY KEY)
                                          (\CHECKNAME RETURN)
                                          (LIST \3BYTEKLUDGEKEY
                                            (COND
                                              (VALIDATEFLG
                                                1)
                                              (T 0])
                                          (FUNCTION \RESPTOSTARTSEND))))
      SENDINGSOCKET)
    (T ; print the reason for failure
      (AND NIL (printout PROMPTWINDOW "Couldn't start sending the message - reason: " STARTSENDRESULT T
        NIL))
      NIL))
```

```
(GV.ADDRECIPIENT
  [LAMBDA (SOCKET NAME) (* M.Yonke "15-JUN-83 15:20")
    (MS.SENDOPERATION \OP.ADDRECIPIENT SOCKET (LIST (\CHECKNAME NAME]))
```

```
(GV.CHECKVALIDITY
  [LAMBDA (SOCKET) (* M.Yonke "15-JUN-83 15:53")
    (MS.SENDOPERATION \OP.CHECKVALIDITY SOCKET NIL (FUNCTION \RESPTOCHECKVAL))
```

```
(GV.STARTITEM
  [LAMBDA (SOCKET TYPE) (* M.Yonke "15-JUN-83 15:31")
    (MS.SENDOPERATION \OP.STARTITEM SOCKET (LIST (OR (AND TYPE (SMALLP TYPE))
      \I.TEXT])
    ; If TYPE is not supplied assume text
```

```
(GV.ADDTOITEM
  [LAMBDA (SOCKET STR) (* bvm%: "24-Jan-86 11:26")
```

;;; Can't use \SENDITEM here because not in usual Grapevine STR format -- no maxLength or padding -- so we do it by hand and no response is given

```
(PROG ((OUTSTREAM (fetch GVOUTSTREAM of SOCKET))
  WASOPEN INSTREAM %#CHARS)
  [SETQ %#CHARS (OR (SELECTQ (TYPENAME STR)
    (STRINGP (NCHARS STR))
    (STREAM (GETFILEINFO [COND
      ((OPENED STR)
        (SETQ WASOPEN (SETQ INSTREAM STR)))
      (T (SETQ INSTREAM (OPENSTREAM STR 'INPUT)
        'LENGTH))
      (LITATOM [COND
        ((INFILEP STR)
          (GETFILEINFO (SETQ INSTREAM (OPENSTREAM STR 'INPUT))
            'LENGTH))
        NIL)
      (NCHARS (SETQ STR (MKSTRING STR]
    (COND
      ((AND INSTREAM (NEQ (GETFILEPTR INSTREAM)
        0))
        (SETFILEPTR INSTREAM 0)))
      (MS.SENDOPERATION \OP.ADDTOITEM SOCKET)
      (while (> %#CHARS MAX.SMALLP) do ; Stream bigger than can be sent in one chunk. Note this cannot be the string case, because
        ; all strings have lengths le MAX.SMALLP
        (\WOUT OUTSTREAM MAX.SMALLP)
        (COPYBYTES INSTREAM OUTSTREAM MAX.SMALLP)
        (SETQ %#CHARS (- %#CHARS MAX.SMALLP))
        (MS.SENDOPERATION \OP.ADDTOITEM SOCKET))
      (\WOUT OUTSTREAM %#CHARS)
      (COND
        (INSTREAM (COPYBYTES INSTREAM OUTSTREAM)
          (OR WASOPEN (CLOSEF INSTREAM)))
        (T (for CHAR instring STR do (BOUT OUTSTREAM CHAR))
```

```
(GV.SEND
  [LAMBDA (SOCKET) (* bvm%: "23-Mar-84 12:42")
    (MS.SENDOPERATION \OP.SEND SOCKET NIL (FUNCTION \RECEIVEACK))
```

```
(MS.EXPAND
  [LAMBDA (SOCKET NAME) (* M.Yonke "15-JUN-83 15:53")
```

;;; Does the mailserver Expand operation -- named to avoid conflict with the database version -- DBEXPAND

```
(MS.SENDOPERATION \OP.MSEXPA ND SOCKET (LIST (\CHECKNAME NAME))
(FUNCTION \RESPTOEXPAND])
```

)

;; Internal Sending

```
(DEFINEQ
```

```
(MS.SENDOPERATION
```

```
[LAMBDA (OP SOCKET ARGS RESPONSEFN)
```

; Edited 3-Sep-87 18:09 by bvm:

;;; basic workhorse for communicating with a mail server --- sends an OP and ARGS and fields a response, if appropriate

```
(COND
[SOCKET (COND
([NLSETQ (LET ((STREAM (fetch GVOUTSTREAM of SOCKET)))
(\WOUT STREAM OP)
(for ARG in ARGS do (\SENDITEM STREAM ARG]
(COND
[RESPONSEFN (CAR (NLSETQ (PROGN (FORCEOUTPUT (fetch GVOUTSTREAM of SOCKET))
(CL:FUNCALL RESPONSEFN (fetch GVINSTREAM of SOCKET]
(T T]
(T
EC.STREAMLOST]) ; We're in the middle --- nothing for it but to bail out
```

```
(\FINDMAILSERVER
```

```
[LAMBDA (ERRORFLG)
```

(\* M.Yonke "15-JUN-83 15:16")

;;; Open a BSP connection to a nearby, responsive mail server and returns it

```
(if (OPENCLOSESTOCKET (\MAILSERVERSOCKETS ERRORFLG)
\MAILSERVERPOLLINGSOC \MAILSERVERENQUIRYSOC NIL \MAILIOTIMEOUT)
elseif ERRORFLG
then (ERROR "Couldn't open connection for" \MAILSERVERNAME)
NIL])
```

```
(\MAILSERVERSOCKETS
```

```
[LAMBDA (ERRORFLG)
```

(\* bvm%: "21-MAY-83 20:00")

```
(PROG (SOCKETS)
(RETURN (COND
([AND \MAILSERVERSOCKETCACHE (NOT (TIMEREXPIRED? (CAR \MAILSERVERSOCKETCACHE]
(CDR \MAILSERVERSOCKETCACHE))
(T (SETQ \MAILSERVERSOCKETCACHE (AND (SETQ SOCKETS (LOCATESOCKETS \MAILSERVERNAME ERRORFLG)
)
(CONS (SETUPTIMER \MAILSOCKETTIMEOUT)
SOCKETS))
SOCKETS])
```

```
(\RECEIVEACK
```

```
[LAMBDA (STREAM)
```

(\* M.Yonke " 3-JUN-83 16:43")

```
(if (BIN STREAM)
then T]) ; any byte will do --- it seems to be 0 when I've noticed
```

```
(\RESPTOCHECKVAL
```

```
[LAMBDA (INSTREAM)
```

(\* bvm%: "24-Jan-86 11:29")

;; As per documentation --- bad guys followed by count of good guys which I CONS on the front

```
(bind N until (EQ (SETQ N (\WIN INSTREAM))
0)
collect (CONS N (\RECEIVERNAME INSTREAM)) finally (RETURN (CONS (\WIN INSTREAM)
$$VAL])
```

```
(\RESPTOEXPAND
```

```
[LAMBDA (INSTREAM)
```

(\* bvm%: "11-MAY-83 16:11")

;; As per documentation --- names followed by a code which I interpret and CONS on the front

```
(while (\RECEIVEBOOL INSTREAM) collect (\RECEIVERNAME INSTREAM)
finally (RETURN (CONS (SELECTQ (BIN INSTREAM)
((0 2)
T)
((1 3)
EC.BADRNAME)
(SHOULDNT))
$$VAL])
```

```
(\RESPTOSTARTSEND
```

```
[LAMBDA (INSTREAM)
```

(\* M.Yonke "26-MAY-83 10:45")

```

(SELECTC (BIN INSTREAM)
  (\RC.SENDSTARTED
    T)
  (\RC.PASSWORDINVALID
    'InvalidPassword)
  (\RC.SENDERNOTREGISTERED
    'SenderNotRegistered)
  (\RC.RETURNTONOTREGISTERED
    'ReturnToNotRegistered)
  (\RC.COMMUNICATIONFAILURE
    'NetworkCommunicationsFailure)
(SHOULDNT])

)

(RPAQ? \MAILIOTIMEOUT NIL)

(RPAQ? \MAILSERVERENQUIRYSOC 46)

(RPAQ? \MAILSERVERNAME ' (Maildrop . ms))

(RPAQ? \MAILSERVERPOLLINGSOC 44)

(RPAQ? \MAILSERVERSOCKETCACHE )

(RPAQ? \MAILSERVERRETRIEVALSOC 47)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \MAILIOTIMEOUT \MAILSERVERENQUIRYSOC \MAILSERVERNAME \MAILSERVERPOLLINGSOC \MAILSERVERSOCKETCACHE
  \MAILSERVERRETRIEVALSOC)

)

(ADDTovar \SYSTEMCACHEVARS \MAILSERVERSOCKETCACHE)

;; Receiving mail

(DEFINEQ

(GV.PORTFROMNAME
  [LAMBDA (SERVERNAME)
    (AND (SETQ SERVERNAME (GV.READCONNECT SERVERNAME))
      (ETHERPORT SERVERNAME))]

(* bvm%: "1-Jan-84 17:11")

(GV.POLLNEWMAIL
  [LAMBDA (GVPORT REGISTEREDNAME)
    (PROG ((SOC (\GETMISC SOCKET))
      (OUTPUP (ALLOCATE.PUP))
      (RESULT '?))
      INPUP)
    (SETUPPUP OUTPUP (fetch (MAILPORT HOST#) of GVPORT)
      (fetch (MAILPORT SOCKET#) of GVPORT)
      \PT.LAURELCHECK NIL SOC T)
    (PUTPUPSTRING OUTPUP REGISTEREDNAME)
    (to \MAXETHERTRIES when (SETQ INPUP (EXCHANGEPUPS SOC OUTPUP NIL T))
      do (SELECTC (fetch PUPTYPE of INPUP)
        (\PT.NEWMAIL (SETQ RESULT T)
          (RETURN))
        (\PT.NONNEWMAIL
          (SETQ RESULT NIL)
          (RETURN))
        (\PT.NOMAILBOX
          (RETURN))
        (\PT.ERROR (AND PUPTRACEFLG (PRINTERORPUP INPUP PUPTRACEFILE))
          (COND
            ((EQ (fetch ERRORPUPCODE of INPUP)
              \PUPE.NOSOCKET)
              (RETURN))))
          NIL)
      finally (AND PUPTRACEFLG (printout PUPTRACEFILE "Mail check timed out" T)))
    (AND INPUP (RELEASE.PUP INPUP))
    (RELEASE.PUP OUTPUP)
    (RETURN RESULT])

(* bvm%: "14-Nov-84 10:10")

(GV.OPENMAILBOX
  [LAMBDA (GVPORT REGISTEREDNAME PASSWORD MAILSERVER)
    (SELECTQ (GV.POLLNEWMAIL GVPORT REGISTEREDNAME)
      (NIL 'EMPTY)
      (? NIL)
      (LET (MAILBOX INBOXRESULT)
        (COND
          ([AND (SETQ MAILBOX (\CONNECTTOMAILSERVER GVPORT))
            (SETQ INBOXRESULT (CAR (NLSETQ (MS.RETRIEVEOPERATION \OP.OPENINBOX MAILBOX
              (LIST (\CHECKNAME REGISTEREDNAME)
                (\CHECKKEY PASSWORD))

```

```

(COND
  ((SMALLP INBOXRESULT)
   (create OPENEDMAILBOX
            MAILBOX _ MAILBOX
            PROPERTIES _ (LIST '%#OFMESSAGES INBOXRESULT)))
  (T
   (create OPENEDMAILBOX
            MAILBOX _ NIL
            PROPERTIES _ INBOXRESULT))
  )

```

**(GV.NEXTMESSAGE**

[LAMBDA (MAILBOX)

(\* DECLARATIONS%: (RECORD  
 (ANOTHERMESSAGE? ARCHIVED? DELETED?)))  
 (\* bvm%: " 5-Nov-84 13:13")

```

(LET (RESULT)
  (SETQ RESULT (MS.RETRIEVEOPERATION \OP.NEXTMESSAGE MAILBOX NIL (FUNCTION \RESPTONEXTMESSAGE)))
  (AND (fetch ANOTHERMESSAGE? of RESULT)
       (LIST 'DELETED (fetch DELETED? of RESULT)
              'ARCHIVED
              (fetch ARCHIVED? of RESULT)))

```

**(GV.RETRIEVEMESSAGE**

[LAMBDA (MAILBOX MSGOUTFILE)

; Edited 6-May-88 16:32 by bvm

```

(LET [(*MSGOUTSTREAM* (GETSTREAM MSGOUTFILE 'OUTPUT]
  (MS.RETRIEVEOPERATION \OP.READMESSAGE MAILBOX NIL (FUNCTION \RESPTORETRIEVEMESSAGE]))

```

**(GV.CLOSEMAILBOX**

[LAMBDA (MAILBOX FLUSHP)

(\* bvm%: " 9-May-84 14:13")

```

(COND
  ((BSPOPENP (fetch GVINSTREAM of MAILBOX))
   (PROG1 [COND
            (FLUSHP (MS.RETRIEVEOPERATION \OP.FLUSH MAILBOX NIL (FUNCTION \RECEIVEACK]
            (CLOSEBSPSTREAM (fetch GVINSTREAM of MAILBOX)
                             \ETHERTIMEOUT))])
  )

```

```

(ADDTOVAR MAILSERVERTYPES (GV GV.POLLNEWMAIL GV.OPENMAILBOX GV.NEXTMESSAGE GV.RETRIEVEMESSAGE GV.CLOSEMAILBOX
  GV.PORTFROMNAME))

```

;; Not currently used

(DEFINEQ

**(GV.READTOC**

[LAMBDA (MAILBOX)

(\* M.Yonke "25-MAY-83 14:37")

```

(MS.RETRIEVEOPERATION \OP.READTOC MAILBOX NIL (FUNCTION \RECEIVESTRING])

```

**(GV.WRITETOC**

[LAMBDA (MAILBOX REMARK)

(\* M.Yonke "25-MAY-83 14:37")

```

(MS.RETRIEVEOPERATION \OP.WRITETOC MAILBOX (LIST REMARK)
  (FUNCTION \RECEIVEACK])

```

**(GV.DELETEMESAGE**

[LAMBDA (MAILBOX)

(\* M.Yonke "25-MAY-83 14:37")

```

(MS.RETRIEVEOPERATION \OP.DELETEMESAGE MAILBOX NIL (FUNCTION \RECEIVEACK])

```

;; Internal Receiving

(DEFINEQ

**(MS.RETRIEVEOPERATION**

[LAMBDA (OP MAILBOX ARGS RESPONSEFN)

; Edited 3-Sep-87 18:09 by bvm:

;;; basic workhorse for communicating with a mail server --- sends an OP and ARGS to MAILBOX and fields a response, if appropriate

```

(LET ((OUTSTREAM (fetch GVOUTSTREAM of MAILBOX)))
  (\WOUT OUTSTREAM OP)
  (for E in ARGS do (\SENDITEM OUTSTREAM E))
  (FORCEOUTPUT OUTSTREAM)
  (COND
    (RESPONSEFN (CL:FUNCALL RESPONSEFN (fetch GVINSTREAM of MAILBOX)))
    (T T))

```

**(\CONNECTTOMAILSERVER**

[LAMBDA (PORT)

(\* bvm%: "24-Feb-86 17:10")

;;; Open a BSP connection to mail server. Its error handler defined as ERROR! so that BAD.STATE.FOR.BIN etc suppressed

```
(RESETVARS ((\RTP.DEFAULTTIMEOUT GV.MAILBOX.TIMEOUT))
;; Crufty!!!! OPENBSPSTREAM should allow RFC timeout to be specified
(RETURN (\OPENVCONNECTION (CONS (CAR PORT)
                                \MAILSERVERRETRIEVALSOC)
      NIL
      (FUNCTION ERROR!]))
```

## (\RESPTOOPENMAILBOX

```
[LAMBDA (INSTREAM)                                     (* bvm%: " 5-Nov-84 16:29")
  (SELECTC (\BIN INSTREAM)
    (\RC.NAMEANDPASSWORDVALID
      (\WIN INSTREAM))
    (\RC.BADPASSWORD
      'BadPassword)
  (PROGN ;; There are actually 5 values for the return code, but most of them are impossible, since Lafite has authenticated NAME
    NIL))
```

## (\RESPTONEXTMESSAGE

```
[LAMBDA (INSTREAM)                                     (* bvm%: "11-MAY-83 15:55")
  (LIST (\RECEIVEBOOL INSTREAM)
        (\RECEIVEBOOL INSTREAM)
        (\RECEIVEBOOL INSTREAM))
```

## (\RESPTORETRIEVEMESSAGE

```
[LAMBDA (INSTREAM)                                     ; Edited 3-Jun-88 18:32 by bvm
  (until (\EOF INSTREAM) bind TYPE BYTELEN GVHOST TIME SENDER ADDEDLENGTH
    do ;; Read an item. Ignore the ones not of type text or tedit formatting
      (SETQ TYPE (\WIN INSTREAM)) ; Item type
      (SETQ BYTELEN (\RECEIVELONGWORD INSTREAM)) ; Number of bytes long it is
      [SELECTC TYPE
        (\I.TEXT ; The text body
          (if SENDER ; Print a postmark first
            then (SETQ ADDEDLENGTH (GETFILEPTR *MSGOUTSTREAM*))
                  (PRINTOUT *MSGOUTSTREAM* "GV-Info: " SENDER " at " (GDATE (ALTO.TO.LISP.DATE
                                                                              TIME))
                  " from "
                  (\CACHED.HOST.NAME GVHOST)
                  T)
                  (SETQ ADDEDLENGTH (- (GETFILEPTR *MSGOUTSTREAM*)
                                         ADDEDLENGTH))
                  ; May have to account for this later
                  (SETQ SENDER NIL))
            (to BYTELEN do ; Use \OUTCHAR to account for eol conventions
              (\OUTCHAR *MSGOUTSTREAM* (BIN INSTREAM)))
              ((CONS \I.TEDITFORMATTING (MKLIST \I.OLDTEDITFORMATTING))
                ; Various TEdit formatting
                (if ADDEDLENGTH
                  then ; We prepended some text, so have to munge the formatting
                        ; instead of copying it straight
                        (LET [(BUFFER (OPENSTREAM ' {NODIRCORE} 'BOTH)
                          (COPYBYTES INSTREAM BUFFER BYTELEN)
                          (LA.ADJUST.FORMATTING BUFFER *MSGOUTSTREAM* ADDEDLENGTH))]
                        ; Just append to the text--the two together make a valid
                        ; textstream
                        (COPYBYTES INSTREAM *MSGOUTSTREAM* BYTELEN)))
                  (OR (AND *GV-SHOW-POSTMARK* (SELECTC TYPE
                    (\I.POSTMARK ; 6 bytes: pupaddr, timelo, timehi
                      (SETQ GVHOST (\WIN INSTREAM))
                      (SETQ TIME (NCREATE 'FIXP))
                      (replace LOWORD of TIME with (\WIN INSTREAM))
                      (replace HIWORD of TIME with (\WIN INSTREAM))
                      T)
                      (\I.SENDER (SETQ SENDER (\RECEIVESTRING INSTREAM (\WIN INSTREAM))
                        ))
                      NIL))
                    (to BYTELEN do (BIN INSTREAM]
                  (COND
                    ((ODDP BYTELEN) ; Eat padding byte
                     (BIN INSTREAM)))
                  finally (BSPGETMARK INSTREAM]))
```

## (\RECEIVELONGWORD

```
[LAMBDA (STREAM)                                     (* bvm%: "11-MAY-83 14:49")
  ;; Read a 32-bit number, low-word is first
  (PROG ((LO (\WIN STREAM))
        (HI (\WIN STREAM)))
```

(RETURN (\MAKENUMBER HI LO))

## (\CACHED.HOST.NAME

[LAMBDA (HOST#)

; Edited 6-May-88 17:25 by bvm

;; Return a name for HOST#. Use cache to avoid recomputation

(OR (CDR (ASSOC HOST# \CACHED.HOST.NAMES))

(CDAR (**push** \CACHED.HOST.NAMES (CONS HOST# (ETHERHOSTNAME HOST# T))

)

(RPAQ? **GV.MAILBOX.TIMEOUT** 12000)(RPAQ? **\*GV-SHOW-POSTMARK\*** )

(RPAQ? \CACHED.HOST.NAMES )

(ADDTOVAR \SYSTEMCACHEVARS \CACHED.HOST.NAMES)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS GV.MAILBOX.TIMEOUT \CACHED.HOST.NAMES)

)

;; LAFITEMODE GV

(ADDTOVAR **LAFITEMODELST** (GV 2 \GV.SEND.PARSE \GV.SENDMESSAGE GV.MAKEANSWERFORM GV.INIT.MAIL.USER \GV.MESSAGE.P  
\GV.MESSAGE.FROM.SELF.P)  
(GRAPEVINE . GV))

(DEFINEQ

## (\GV.INIT.MAIL.USER

[LAMBDA NIL

; Edited 9-Nov-89 15:31 by bvm

(LET\* ((GVUSERNAME (LAFITE.USER.NAME.FROM.LOGIN T T))

(FULLNAME (CONCAT (CAR GVUSERNAME)

"."

(CDR GVUSERNAME)))

PASS MAILSERVERS AUTHENTICATED?)

(COND

((NEQ [SETQ AUTHENTICATED? (GV.AUTHENTICATE GVUSERNAME (SETQ PASS (GV.MAKEKEY (CDR (  
\INTERNAL/GETPASSWORD  
])

T)

(printout PROMPTWINDOW T "Cannot authenticate user " FULLNAME " because: " (SETQ  
\LAFITE.AUTHENTICATION.FAILURE  
AUTHENTICATED?)

".")

NIL)

([NULL (SETQ MAILSERVERS (CDR (GV.EXPAND GVUSERNAME]

(printout PROMPTWINDOW T "There are no mail servers for user " FULLNAME)

NIL)

(T (**create** LAFITEMODEDATA

FULLUSERNAME \_ FULLNAME

UNPACKEDUSERNAME \_ GVUSERNAME

CREDENTIALS \_ PASS

SHORTUSERNAME \_ FULLNAME

FROMFIELD \_ (AND LAFITE.GV.FROM.FIELD (LET ((PARSE (\GV.PARSERECIPIENTS1  
LAFITE.GV.FROM.FIELD NIL T)));; If the user's variable parses correctly into the authenticated  
;; user, then take it(if (AND PARSE (NULL (CDR PARSE))  
(STRING-EQUAL (CAAR PARSE)  
(CAR GVUSERNAME))  
(STRING-EQUAL (CDAR PARSE)  
(CDR GVUSERNAME)))**then** LAFITE.GV.FROM.FIELD**else** (CL:FORMAT PROMPTWINDOW "~%%Can't use GV  
From field %"~A%" because it does  
not parse to current user  
%"~A%" LAFITE.GV.FROM.FIELD  
FULLNAME)

NIL)))

MAILSERVERS \_

(for MAILSERVER in MAILSERVERS **bind** SERVEROPS SERVERPORT SERVERDEF**when** [COND((NULL (SETQ SERVEROPS (**GETMAILSERVEROPS** MAILSERVER)))

NIL)

((NULL (SETQ SERVERPORT (CL:FUNCALL (**fetch** (MAILSERVEROPS SERVERPORTFROMNAME)  
**of** SERVEROPS)

MAILSERVER)))

(printout PROMPTWINDOW T "Can't find address of " MAILSERVER)

NIL)

(T (SETQ SERVERDEF (**create** MAILSERVER

MAILPORT \_ SERVERPORT

```
MAILSERVERNAME _ MAILSERVER
MAILSERVEROPS _ SERVEROPS]
```

collect SERVERDEF])

**(GETMAILSERVEROPS**

```
[LAMBDA (MAILSERVER)
  (PROG ((SERVERTYPE (\GV.MAILSERVERTYPE MAILSERVER))
    OPS)
    (RETURN (COND
      ([AND SERVERTYPE (SETQ OPS (OR (ASSOC SERVERTYPE MAILSERVERTYPES)
        (AND (EQ SERVERTYPE 'MTP)
          (PROGN (FILESLOAD (SYSLOAD FROM VALUEOF
            LISPUSERSDIRECTORIES)
              MTP)
            (ASSOC SERVERTYPE MAILSERVERTYPES])
          (CDR OPS))
      (T (printout PROMPTWINDOW T "Lafite cannot retrieve mail from " MAILSERVER)
        NIL]))
```

(\* bvm%: "12-Nov-84 17:52")

**(\GV.MAILSERVERTYPE**

[LAMBDA (MAILSERVERNAME)

(\* bvm%: " 9-Dec-85 17:03")

;;; type is determined by the name currently

```
(COND
  ((STRING-EQUAL (SUBSTRING MAILSERVERNAME -3)
    ".MS")
    'GV)
  ((STRING-EQUAL MAILSERVERNAME "MAXC")
    'MTP])
)
```

(DEFINEQ

**(\GV.SENDMESSAGE**

[LAMBDA (MSG PARSE EDITORWINDOW ABORTWINDOW)

; Edited 15-Jun-90 17:53 by jds

;;; This is the real mail sender for the GrapeVine

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

```
(PROG ((PWINDOW (AND EDITORWINDOW (GETPROMPTWINDOW EDITORWINDOW)))
  (RECIPIENTS (fetch (GVMAILPARSE GVPRECIPIENTS) of PARSE))
  (FROMFIELD (fetch (GVMAILPARSE GVPFROM) of PARSE))
  (FORMATTING (fetch (GVMAILPARSE GVPFORMAT) of PARSE))
  (DATEFIELD (CONCAT "Date: " (DATE (DATEFORMAT NO.SECONDS TIME.ZONE SPACES))
    LAFITEEOL))
  (FAILURECNT 0)
  (SENDER (fetch (LAFITEMODEDATA FULLUSERNAME) of *LAFITE-MODE-DATA*))
  (UNPACKEDSENDER (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of *LAFITE-MODE-DATA*))
  DATELEN SENDINGSOCKET RECIPIENTSCHECK SENDRESULT SENDERFIELD ABORTMENU)
[COND
  ((NOT (TYPENAMEP MSG 'STREAM))
    (RETURN (\ILLEGAL.ARG MSG)
      (SETQ FORMATTING (SELECTQ FORMATTING
        ((MULTIMEDIA TEDIT) ; Send with TEdit formatting, assuming there is any
          (AND (TEDIT.FORMATTEDFILEP MSG)
            T))
        (TEXT NIL)
        (\ILLEGAL.ARG FORMATTING)))
    [COND
      (PWINDOW (CLEARW PWINDOW)
        (CL:FORMAT PWINDOW "Delivering ~:[~;formatted ~]to ~D recipient~:P" FORMATTING (LENGTH
          RECIPIENTS
          ]
      [SETQ SENDERFIELD
        (COND
          (FROMFIELD ; Test for valid From field. We waited til now to do this so we
            ; didn't hold up start of delivery.
            (if (for ADDR in FROMFIELD bind REG unless (STRPOS "@" (CAR ADDR))
              do ; assume arpa addresses are valid
                (if (NLISTP (GV.EXPAND (if (EQ (SETQ REG (CDR ADDR))
                  'NOREGISTRY)
                then ; The address had no registry. We do it this odd way so that we
                  ; can distinguish "Fred" from "Fred.PA" in the test below.
                  (CONS (CAR ADDR)
                    (SETQ REG (CDR UNPACKEDSENDER)))
                else ADDR)))
            then [COND
              (PWINDOW (CLEARW PWINDOW)
                (CL:FORMAT PWINDOW "From field not valid address: ~A"
                  (CONCAT (CAR ADDR)
                    ". " REG)
                  (RETURN T)))
```



```

    then (RETURN NIL))
  (if (AND (NULL (CDR FROMFIELD))
          (STRING-EQUAL (CONCAT (CAAR FROMFIELD)
                                "."
                                (CDAR FROMFIELD))
                          SENDER))
      then
        ; From field is just a fancy way of writing the real from field, so
        ; adding a Sender field would be redundant
        NIL
      else (CONCAT "Sender: " SENDER LAFITEEOL)))
  (T (CONCAT "From: " (OR (fetch (LAFITEMODEDATA FROMFIELD) of *LAFITE-MODE-DATA*)
                          SENDER)
             LAFITEEOL])
[COND
  (FORMATTING (TEDIT.INSERT MSG DATEFIELD 1 NIL T)
    (SETQ DATELEN (NCHARS DATEFIELD))
    (CL:WHEN SENDERFIELD
      (TEDIT.INSERT MSG SENDERFIELD (ADD1 DATELEN)
                    NIL T))
    ; Do tedit conversion now, before we have the bsp stream tied up
    [SETQ MSG (PROG1 (COERCETEXTOBJ MSG 'SPLIT)
                    (TEDIT.DELETE MSG 1 (if SENDERFIELD
                                             then (+ DATELEN (NCHARS SENDERFIELD))
                                             else DATELEN)))])
    (SETQ FORMATTING (CDR MSG))
    (SETQ MSG (CAR MSG)))
  ((AND (TEXTSTREAMP MSG)
        (TEDIT.FORMATTEDFILEP MSG))
    ; Message has formatting, but caller asked to send it as plain
    ; text. Carefully coerce it, since TEDIT ns chars and image
    ; objects don't pass thru COPYBYTES very well
    (SETQ MSG (LAFITE.MAKE.PLAIN.TEXTSTREAM MSG))
STARTSEND
  (as I to 3 until (SETQ SENDINGSOCKET (GV.STARTSEND UNPACKEDSENDER (fetch (LAFITEMODEDATA CREDENTIALS)
                                                                              of *LAFITE-MODE-DATA*)
                                                                              UNPACKEDSENDER T))
                  ; loop 3 times trying to start this send
                  (DISMISS 1000))
[COND
  ((NULL SENDINGSOCKET)
   (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Couldn't connect to a maildrop")
   (RESETSAVE NIL (LIST (FUNCTION GV.KILL SOCKET)
                        SENDINGSOCKET))
   (AND PWINDOW (printout PWINDOW '|...|)))
(COND
  ((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW 'ABORT))
   (ERROR!)))
(SELECTQ (SETQ RECIPIENTSCHECK (\GV.SENDRECIPIENTS SENDINGSOCKET RECIPIENTS EDITORWINDOW))
  (NIL
   ; MS didn't like the recipients list -- this was already reported by
   ; \GV.SENDRECIPIENTS
   (RETURN NIL))
  (?
   ; Something went wrong, try again
   (GO TRYAGAIN))
  NIL)
; Everything is OK
(COND
  ((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW 'ABORT))
   (ERROR!)))
(GV.STARTITEM SENDINGSOCKET)
; send code to start sending text
[COND
  ((NOT FORMATTING)
   (GV.ADDTOITEM SENDINGSOCKET DATEFIELD)
   (AND SENDERFIELD (GV.ADDTOITEM SENDINGSOCKET SENDERFIELD))
   ; send the message
   (COND
    ((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW 'ABORT))
     (ERROR!)))
   (GV.ADDTOITEM SENDINGSOCKET MSG)
   ; tell the grapevine to send the message
   (COND
    (FORMATTING (COND
      ((AND ABORTWINDOW (WINDOWPROP ABORTWINDOW 'ABORT))
       (ERROR!)))
    (GV.STARTITEM SENDINGSOCKET \I.TEDITFORMATTING)
    ; Send formatting info
    (GV.ADDTOITEM SENDINGSOCKET FORMATTING)))
(COND
  ((NULL ABORTWINDOW))
  (WINDOWPROP ABORTWINDOW 'ABORT)
  (ERROR!))
  (T
   ; Too late to abort now
   (DELETEMENU [SETQ ABORTMENU (CAR (WINDOWPROP ABORTWINDOW 'MENU)
                                         NIL ABORTWINDOW)))
(COND
  ((EQ (SETQ SENDRESULT (GV.SEND SENDINGSOCKET))
        T)
   (RETURN T)))
TRYAGAIN
[COND
  ((> (add FAILURECNT 1)
    4)

```

```

    (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW "Several unsuccessful attempts"]
    (AND PWINDOW (printout PWINDOW " problems, trying again.))
    (GV.KILLSOCKET SENDINGSOCKET) ; Just in case it's still alive
    (COND
      (ABORTMENU ; Restore the Abort menu that we took down
        (ADDMENU ABORTMENU ABORTWINDOW '(0 . 0))
        (SETQ ABORTMENU)))
    (GO STARTSEND])

```

## (\GV.SENDRECIPIENTS

```

  [LAMBDA (SOCKET RECIPIENTS EDITORWINDOW)
    (* DECLARATIONS%: (RECORD
      (%#OFVALIDRECIPIENTS . INVALIDRECIPIENTS))
      (RECORD INVALIDRECIPIENT (RECIPIENT#
        . RECIPIENTNAME)
      (* bvm%: " 6-Nov-84 11:53")
    ))
    (PROG (REASON VALIDITYRESULT INVALID)
      [COND
        ((NLISTP RECIPIENTS)
          (SETQ REASON "No recipients supplied"))
        (T (for R in RECIPIENTS do (GV.ADDRECIPIENT SOCKET R))
          (SETQ VALIDITYRESULT (GV.CHECKVALIDITY SOCKET))
          (COND
            ((NLISTP VALIDITYRESULT)
              (RETURN '?))
            ((NULL (SETQ INVALID (fetch INVALIDRECIPIENTS of VALIDITYRESULT)))
              ; everything went OK
              (RETURN VALIDITYRESULT))
            (T ; GV didn't like some recipients
              (SETQ REASON (\LAFITE.INVALID.RECIPIENTS (for RECIPIENT in INVALID
                collect (fetch (INVALIDRECIPIENT RECIPIENTNAME)
                  of RECIPIENT]
                FAILED
                (RETURN (\LAFITE.SEND.FAIL EDITORWINDOW REASON]))
      )
    )
  )

```

(DEFINEQ

## (\GV.SEND.PARSE

```

  [LAMBDA (MSG EDITORWINDOW) ; Edited 5-May-88 13:06 by bvm
    (PROG ((REGISTRY (CDR (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of *LAFITE-MODE-DATA*)))
      RECIPIENTS MSGFIELDS FROMFIELD SENDINGFORMAT HEADEREOF REPLYTO SUBJECT)
      (OR (SETQ MSGFIELDS (\LAFITE.PREPARE.SEND MSG EDITORWINDOW))
        (RETURN))
      [COND
        ((EQ (CAAR MSGFIELDS)
          'EOF)
          (SETQ HEADEREOF (CADR (pop MSGFIELDS)
            (for PAIR in MSGFIELDS do (SELECTQ (CAR PAIR)
              (Date (\SENDMESSAGEFAIL EDITORWINDOW "User-supplied Date not allowed"))
              ((To cc)
                (SETQ RECIPIENTS (NCONC RECIPIENTS (\GV.PARSERECIPIENTS (CDR PAIR)
                  REGISTRY T EDITORWINDOW)))
                (From (SETQ FROMFIELD (\GV.PARSERECIPIENTS (CDR PAIR)
                  'NOREGISTRY T EDITORWINDOW)))
                (Reply-to (SETQ REPLYTO (\GV.PARSERECIPIENTS (CDR PAIR)
                  REGISTRY T EDITORWINDOW)))
                (Sender (\SENDMESSAGEFAIL EDITORWINDOW "User-supplied Sender not
                  allowed"))
                (Format (SETQ SENDINGFORMAT (CADR PAIR)))
                (Subject (SETQ SUBJECT (CADR PAIR)))
                NIL)))
          (COND
            ((NULL RECIPIENTS)
              (\SENDMESSAGEFAIL EDITORWINDOW "No recipients!")
              (RETURN NIL))
            ((FMEMB NIL RECIPIENTS)
              ;; if there is a NIL in RECIPIENTS then \GV.PARSERECIPIENTS couldn't parse something {it already reported it} therefore just get
              ;; out now
              (RETURN NIL)))
          [COND
            ((NULL SENDINGFORMAT)
              (SETQ SENDINGFORMAT (OR (\LAFITE.CHOOSE.MSG.FORMAT MSG HEADEREOF EDITORWINDOW)
                (RETURN))
            [COND
              (([AND EDITORWINDOW (NULL REPLYTO)
                (for GVNAME in RECIPIENTS thereis (EQ (NTHCHARCODE (CAR GVNAME)
                  -1)
                  (CHARCODE ^]
                (OR (\LAFITE.CHOOSE.REPLYTO MSG HEADEREOF FROMFIELD EDITORWINDOW)
                  (RETURN)
                (RETURN (create GVMAILPARSE
                  GVPSUBJECT _ SUBJECT
                  GVFFROM _ FROMFIELD

```

GVPFORMAT \_ SENDINGFORMAT  
GVPRECIPIENTS \_ RECIPIENTS])

## (\GV.PARSERECIPIENTS

[LAMBDA (FIELD REGISTRY INTERNALFLG EDITWINDOW)  
[SETQ FIELD (COND

; Edited 9-Sep-87 16:27 by bvm:

((LISTP FIELD)  
(for PIECE in FIELD join (\GV.PARSERECIPIENTS1 PIECE REGISTRY INTERNALFLG EDITWINDOW)))  
(T (\GV.PARSERECIPIENTS1 FIELD REGISTRY INTERNALFLG EDITWINDOW]  
(COND  
(INTERNALFLG FIELD)  
(T (CL:REMOVE-DUPLICATES FIELD :TEST 'STRING-EQUAL]))

## (\GV.PARSERECIPIENTS1

[LAMBDA (LINE REGISTRY INTERNALFLG EDITWINDOW RECURSIVE-P)

; Edited 13-Jun-88 12:21 by bvm

;;; INTERNALFLG = T means produce addresses to give Grapevine; NIL means give human-readable addresses

```
(PROG ((I 0)
  NC NOPEN CH ADDRESSES ADDR FAILURE OPEN CLOSE DOT ATSIGN START COLON COMSTART COMMENTS PRETTY DLNAME)
  (COND
    ( (NULL LINE)
      (RETURN NIL)))
    (SETQ NC (NCHARS LINE))
  NEXTADDR
    (if (>= I NC)
      then
        (if COLON
          then
            (SETQ FAILURE #\:)
            (GO FAIL))
          (RETURN (REVERSE ADDRESSES)))
        (CASE (SETQ CH (CL:CHAR LINE I))
          ((#\Space #\Tab #\Newline)
            (add I 1)
            (GO NEXTADDR)))
          (SETQ START I)
          [SETQ OPEN (SETQ CLOSE (SETQ DOT (SETQ ATSIGN (SETQ COMMENTS NIL)
THISCHAR
          (CASE CH
            (#\" (GO INQUOTE))
            (#\" (GO INCOMMENT))
            (#\" (GO INDOMAIN))
            ((#\ ) #\))
              (SETQ FAILURE CH)
              (GO FAIL))
            (#\"<
              (if OPEN
                then (SETQ FAILURE #\"<)
                (GO FAIL)
              else (SETQ OPEN I)))
            (#\">
              (if OPEN
                then (SETQ CLOSE I)
                else (SETQ FAILURE #\">)
                (GO FAIL)))
            (#\". (if (NOT CLOSE)
              then
                (SETQ DOT I)))
            (#\"@ (if (NOT CLOSE)
              then
                (SETQ ATSIGN I)
                (SETQ DOT NIL))
            (#\": (if (OR OPEN ATSIGN COLON)
              then (SETQ FAILURE "Invalid use of colon")
              (GO FAIL)
            else (if (EQ (\GV.FIND.NON.SPACE LINE (+ I 1)
              NC)
                #\";)
              then
                (SETQ ADDR (\GV.HANDLE.DL (SETQ DLNAME (\GV.EXTRACT.FIELD LINE START I)
INTERNALFLG EDITWINDOW))
                (if (EQ (CAR ADDR)
                  :ERROR)
                  then (SETQ FAILURE (CADR ADDR))
                  (GO FAIL)
                else (SETQ ADDRESSES (NCONC ADDR ADDRESSES))
                  (SETQ DLNAME NIL))
              else
                (SETQ COLON I)
                (add I 1)
                (GO NEXTADDR)))
```

; done

; Slight missing semi-colon

; Ignore leading whitespace

; Unbalanced stuff

; Start of address spec

; End of address spec

; Note placement of periods. Ignore if outside &lt;&gt; address.

; Note arpa address separator

; DL in the form "dlname:." -- get recipients from file

; Random phrase naming the group, followed by the  
; addresses--just parse them as they come along, and note that  
; we expect a semi-colon at some point.

```

(NIL #\, #\;) ; end of address
(if (CASE CH
    (#\; [if (NOT COLON)
              then (SETQ FAILURE #\;)
              else (CASE (\GV.FIND.NON.SPACE LINE (+ I 1)
                        NC)
                    ((NIL #\,) ; good--semi is at end of address. Consider it eaten
                     (SETQ COLON NIL))
                    (T
                     (SETQ FAILURE "Semi-colon must be at end of group specification")))]))
    ((NIL) ; end of everything, check that there's no outstanding colon
     (if COLON
         then (SETQ FAILURE #\:)))
    then (GO FAIL)) ; Problem with this semi-colon
(if (NEQ START I)
    then ; there is a name
        [if (NOT OPEN)
            then ; simple address
                (SETQ ADDR (\GV.PARSE.SIMPLE.RECIPIENT LINE START I DOT ATSIGN COMMENTS
                    REGISTRY INTERNALFLG))
                (if (EQ (CAR ADDR)
                        :ERROR)
                    then (SETQ FAILURE (CADR ADDR))
                        (GO FAIL))
                elseif (NOT CLOSE)
                    then (SETQ FAILURE #\<)
                        (GO FAIL)
                else ; real address is inside the <>
                    (SETQ ADDR (\GV.PARSE.SIMPLE.RECIPIENT LINE (+ OPEN 1)
                        CLOSE DOT ATSIGN COMMENTS REGISTRY INTERNALFLG))
                    (if (EQ (CAR ADDR)
                            :ERROR)
                        then (SETQ FAILURE (CADR ADDR))
                            (GO FAIL))
                    (CASE INTERNALFLG
                        ((NIL :BOTH) ; Want pretty address--stick the reparsed real address inside the
                                    ; template
                                     (SETQ PRETTY (CL:FORMAT NIL "~@[~A ~]<~A>~@[ ~A~]" (\GV.EXTRACT.FIELD
                                         LINE START OPEN)
                                         (if INTERNALFLG
                                             then (CAR ADDR)
                                             else ADDR)
                                         (\GV.EXTRACT.FIELD LINE (+ CLOSE 1)
                                         I)))
                                     (SETQ ADDR (if INTERNALFLG
                                         then ; Want (pretty . internal)
                                             (CONS PRETTY (CDR ADDR))
                                         else PRETTY))))])
                    (push ADDRESSES ADDR))
                (add I 1)
                (GO NEXTADDR)))
NEXT-I
(SETQ CH (AND (< (add I 1)
                NC)
              (CL:CHAR LINE I)))
(GO THISCHAR)
INQUOTE

;; Parse a quoted string--skip to next quote
[while (< (add I 1)
          NC)
  do (CASE (CL:CHAR LINE I)
      (#\" ; end of quoted text
        (GO NEXT-I))
      (#\\ ; quotes next char
        (add I 1)))])

;; If we get here, we have an unbalanced quote
(SETQ FAILURE #\" )
(GO FAIL)
INCOMMENT

;; Parse a comment in parens. Parentheses may be nested. Add to set of comments
(SETQ COMSTART I)
(SETQ NOPEN 1)
[while (< (add I 1)
          NC)
  do (CASE (CL:CHAR LINE I)
      (#\ (if (EQ (SETQ NOPEN (- NOPEN 1))
                  0)
              then ; end of comment--parens are now matched
                  [if INTERNALFLG
                      then ; We will need to know how to strip comments
                          (push COMMENTS (LIST COMSTART (+ I 1))

```

```

                (GO NEXT-I)))
        (#\ (
            (add NOPEN 1))
        (#\ (
            (add I 1)))])
;; If we get here, we have an unbalanced paren
    (SETQ FAILURE #\()
    (GO FAIL)
INDOMAIN

;; Parse a domain literal--string in square brackets. These may not be nested.
[while (< (add I 1)
        NC)
  do (CASE (CL:CHAR LINE I)
        (#\
            (GO NEXT-I))
        (#\[
            (RETURN))
        (#\ (
            (add I 1)))])
;; If we get here, we have an unbalanced bracket
    (SETQ FAILURE #\[)
    (GO FAIL)
FAIL
    (RETURN (COND
        (RECURSIVE-P (LIST :ERROR FAILURE))
        ((NOT EDITWINDOW)
            (LIST NIL))
        [(CL:CHARACTERP FAILURE)
            (\SENDMESSAGEFAIL EDITWINDOW (CL:FORMAT NIL "Error~@[ in ~A~]: " DLNAME)
                (CASE FAILURE
                    (#\ "Unmatched quotation mark")
                    (#\ "Incorrect group syntax--colon without terminating semi-colon")
                    (T (CL:FORMAT NIL "Unmatched %~C%" FAILURE))))]
        (DLNAME (\SENDMESSAGEFAIL EDITWINDOW (CONCAT "In " DLNAME ": " )
            FAILURE))
        (T (\SENDMESSAGEFAIL EDITWINDOW FAILURE]))
        ; No interaction, just show failure by returning (NIL)

```

## (\GV.FIND.NON.SPACE

[LAMBDA (STR START END)

; Edited 10-Jun-88 17:11 by bvm

;; returns the next non-white-space char in str from position start to end.

```

(bind CH do (if (>= START END)
                then (RETURN NIL)
                else (CASE (SETQ CH (CL:CHAR STR START))
                    ((#\Space #\Tab #\Newline) (add START 1))
                    (T (RETURN CH))))))

```

## (\GV.PARSE.SIMPLE.RECIPIENT

[LAMBDA (FIELD START END DOT ATSIGN COMMENTS REGISTRY INTERNALFLG)

; Edited 10-Jun-88 17:11 by bvm

;;; Parses a single ADDRESS, a list, and returns a proper address as a string, or if INTERNALFLG, in the form Grapevine likes

```

(COND
  [(EQ INTERNALFLG :BOTH)
    (LET ((INTERNAL (\GV.PARSE.SIMPLE.RECIPIENT FIELD START END DOT ATSIGN COMMENTS REGISTRY T)))
      (AND INTERNAL (CONS (\GV.PARSE.SIMPLE.RECIPIENT FIELD START END DOT ATSIGN COMMENTS REGISTRY NIL)
                          INTERNAL))
      [INTERNALFLG (LET (TMP)
                      (if (NULL DOT)
                          then (SETQ TMP (\GV.EXTRACT.FIELD FIELD START END COMMENTS))
                          (if (NULL ATSIGN)
                              then
                                  ; use default registry
                                  (if REGISTRY
                                      then (CONS TMP REGISTRY)
                                      else (LIST :ERROR (CONCAT "No registry given for addressee " TMP)))
                              elseif (OR (EQ (CL:CHAR TMP 0)
                                           #\@)
                                      (EQ (CL:CHAR TMP (- (NCHARS TMP)
                                                           1))
                                           #\@))
                                  then
                                      ; @ but null name or domain?
                                      (\GV.PARSE.FAILED TMP)
                                  else
                                      ; Assume name is otherwise good, use default arpa registry
                                      (CONS TMP ARPANETGATEWAY.REGISTRY))
                      (SETQ TMP (CONS (\GV.EXTRACT.FIELD FIELD START DOT COMMENTS)
                                      (\GV.EXTRACT.FIELD FIELD (+ DOT 1)
                                                             END COMMENTS)))
                      (if (NULL (CDR TMP))
                          then
                              ; no domain?

```

```

(\GV.PARSE.FAILED (CONCAT (CAR TMP)
                          ".")
  else [RPLACD TMP (MKATOM (U-CASE (CDR TMP)
                                   ; Grapevine registry must be atom, sigh.
                                   TMP]
        (OR ATSIGN DOT)
        ; have an ARPA Internet address, or Grapevine address with
        ; registry--nothing to add to make user-sensible address
        (if (NULL (\GV.FIND.NON.SPACE FIELD (OR ATSIGN DOT)
                                                END))
            then
              (\GV.PARSE.FAILED (\GV.EXTRACT.FIELD FIELD START END)) ; no domain
            else (\GV.EXTRACT.FIELD FIELD START END))
        (REGISTRY
          (CONCAT (\GV.EXTRACT.FIELD FIELD START END)
                  ". " REGISTRY)) ; Address without registry, supply default
        (T
          (\GV.EXTRACT.FIELD FIELD START END)) ; Not even a default registry? Punt

```

## (\GV.EXTRACT.FIELD

; Edited 10-Jun-88 17:11 by bvm

```

[LAMBDA (STR START END COMMENTS)
  (if COMMENTS
    then (LET (TMP PIECES)
            (SETQ PIECES (for PAIR in (if (CDR COMMENTS)
                                          then (REVERSE COMMENTS)
                                          else COMMENTS)
              when [AND (<= START (CAR PAIR))
                       (< (CAR PAIR)
                           END)
                (PROG1 (SETQ TMP (\GV.EXTRACT.FIELD STR START (CAR PAIR)))
                        (SETQ START (CADR PAIR)))]
              join (LIST " " TMP)))
          [if (SETQ TMP (\GV.EXTRACT.FIELD STR START END))
            then (SETQ PIECES (NCONC PIECES (LIST " " TMP)
                                                (CONCATLIST (CDR PIECES)))]
          else
            (while (AND (< START END)
                        (CASE (CL:CHAR STR START)
                            ((#\Space #\Tab #\Newline) T)))
              do (add START 1))
            (while (AND (< START END)
                        (CASE (CL:CHAR STR (- END 1))
                            ((#\Space #\Tab #\Newline) T)))
              do (add END -1))
            (AND (< START END)
                 (CL:SUBSEQ STR START END]))
            ; trim white space from edges

```

## (\GV.HANDLE.DL

; Edited 13-Jun-88 14:17 by bvm

```

[LAMBDA (DL INTERNALFLG EDITWINDOW)
  (CASE INTERNALFLG
    (NIL :BOTH)
    ; Don't want the actual addresses, just something to stick in a
    ; header. We assume user of :BOTH is answer, or something
    ; that is mainly concerned with textual presentation to user.

    (LET ((STR (CONCAT DL ";;"))
          (LIST (if INTERNALFLG
                    then
                      (LIST* STR STR NIL)
                    else STR))))
      (T [LET ((FILENAME (PACKFILENAME.STRING 'BODY (if (EQL (CL:CHAR DL 0)
                                                            #\")
                  then ; quoted file name, take off the quotes first
                      (CL:SUBSEQ DL 1 (- (CL:LENGTH DL)
                                          1))
                  else DL)
              'EXTENSION LAFITEDL.EXT))
          STREAM)
        (if [NULL (SETQ FILENAME (if (OR (UNPACKFILENAME.STRING FILENAME 'HOST)
                                         (UNPACKFILENAME.STRING FILENAME 'DIRECTORY))
              then (INFILEP FILENAME)
              else ; Search default directories
                  (FINDFILE FILENAME T (CONS LAFITEDEFAULTHOST&DIR
                                              LAFITEDLDIRECTORIES))
              then (LIST :ERROR "Can't find file by this name")
            elseif [NULL (SETQ STREAM (CAR (NLSETQ (OPENTEXTSTREAM (MKATOM FILENAME)
              then (LIST :ERROR (CONCAT "Can't open " DL))
            else (RESETLIST
                  (RETSAVE NIL (LIST 'CLOSEF STREAM)) ; I hope this closes the file. We used OPENTEXTSTREAM
                  ; instead of OPEN so that file can contain tedit formatting.
                  (bind LINE SOME RESULT while (SETQ LINE (CL:READ-LINE STREAM NIL NIL))
                    when (SETQ SOME (\GV.PARSERECIPIENTS1 LINE NIL INTERNALFLG EDITWINDOW T))
                    do ; Note that we parse with respect to NO registry--require that the
                      ; file contain all complete addresses
                      (if (EQ (CAR SOME)
                          :ERROR)
                        then (RETURN SOME)
                        else (SETQ RESULT (NCONC SOME RESULT))))

```

**finally** ; Whole file parsed ok, so return the result  
 (RETURN RESULT)))]))]

## (\GV.PARSE.FAILED

[LAMBDA (ADDRESS) ; Edited 13-Oct-87 14:17 by bvm:  
 (LIST :ERROR (CONCAT "Bad addressee " ADDRESS))]

## (\LAFITE.CHOOSEREPLYTO

[LAMBDA (TEXTSTREAM HEADEREOF FROMFIELD EDITORWINDOW) ; Edited 3-Nov-89 12:59 by bvm

;;; Invoked when sending to a distribution list. Ask user for Reply-to: field, and if one is chosen, enter it into message

```
(COND
  ((NULL EDITORWINDOW)
   T)
  (T (SELECTQ (\SENDMESSAGE.MENUPROMPT EDITORWINDOW (OR LAFITEREPLYTOMENU (SETQ LAFITEREPLYTOMENU
                                                                                   (\LAFITE.CREATE.MENU
                                                                                   LAFITEREPLYTOMENUITEMS
                                                                                   "Include a Reply-to
                                                                                   field?" T)))
               "Message is addressed to a distribution list"
               'LAFITEREPLYTOMENU)
    (NO T)
    (SELF (\LAFITE.INSERT.REPLYTO TEXTSTREAM
      (COND
        [FROMFIELD ; Message explicitly from someone other than logged-in user, so
                    ; set accordingly
          (CONCATLIST (CDR (for GVNAME in FROMFIELD
                           join (CONS ", " (COND
                                ((EQ (CDR GVNAME)
                                     ARPANETGATEWAY.REGISTRY)
                                 (LIST (CAR GVNAME)))
                                (T (LIST (CAR GVNAME)
                                         ". "
                                         (CDR GVNAME))
                                   (CDR GVNAME)
                                   (T (fetch (LAFITEMODEDATA FULLUSERNAME) of *LAFITE-MODE-DATA*))
                                   NIL HEADEREOF)
                                T)
              (OTHER (\LAFITE.INSERT.REPLYTO TEXTSTREAM ">>Address<<" T HEADEREOF)
                NIL)
              (ABORT NIL)
              NIL))
      )
    )
  )
```

(DEFINEQ

## (\GV.MESSAGE.P

[LAMBDA (MSG) ; Edited 6-May-88 13:58 by bvm

;; Test whether the specified message is a GV message. We test for sender having gv format. This may fail for msgs with malformed from, but  
 ;; that's too bad.

```
(LET* ((SENDER (fetch (LAFITEMSG FROM) of MSG))
       (DOT (STRPOS "." SENDER)))
  (if (NULL DOT)
      then ; No registry
      NIL
      elseif (NULL (STRPOS ":" SENDER))
      then ; Registry and no colon, so believe it. This is based on the
            ; assumption that all grapevine users are internal Xerox with only
            ; NS as competition.
      T
      else ; Subject it to the rigorous parser.
        (NOT (FMEMB NIL (\GV.PARSERECIPIENTS1 SENDER))
```

## (\GV.MESSAGE.FROM.SELF.P

[LAMBDA (MSG) ; Edited 6-May-88 14:36 by bvm

;; True if message is from current user

```
(LET ((SENDER (fetch (LAFITEMSG FROM) of MSG))
      (ME (fetch (LAFITEMODEDATA FULLUSERNAME) of *LAFITE-MODE-DATA*))
      HIT)
  (AND (SETQ HIT (STRPOS ME SENDER NIL NIL NIL NIL UPPERARRAY))
       (if (EQ (SETQ HIT (SUB1 HIT))
               0)
           then ; Matched at start--is the length right?
             (EQ (NCHARS ME)
                  (NCHARS SENDER))
           elseif (EQ (CL:CHAR SENDER (SUB1 HIT))
                     #\<)
           then ; Also recognize "name <realname>"
             (AND (< (add HIT (NCHARS ME))
                      (NCHARS SENDER))
```

```
(EQ (CL:CHAR SENDER HIT)
    #\>])
```

**(GV.MAKEANSWERFORM**

```
[LAMBDA (MSGDESCRIPTORS MAILFOLDER) ; Edited 10-Jun-88 17:27 by bvm
  (LET (MSGFIELDS (\LAFITE.PARSE.MESSAGE MAILFOLDER (OR (CAR (LISTP MSGDESCRIPTORS))
    MSGDESCRIPTORS)))
    SUBJECT FROM DATE SENDER REPLYTO TO CC ORIGINALREGISTRY OLDFROM NEWTO NEWCC)
    ; get the fields from the file
    (for PAIR in MSGFIELDS do (SELECTQ (CAR PAIR)
      (Subject (SETQ SUBJECT (CADR PAIR)))
      (Sender (SETQ SENDER (CADR PAIR)))
      (From (SETQ FROM (CADR PAIR)))
      (Date (SETQ DATE (CADR PAIR)))
      (Reply-to (SETQ REPLYTO (CADR PAIR)))
      (To (SETQ TO (CADR PAIR)))
      (cc (SETQ CC (CADR PAIR)))
      NIL))
    ;; first parse the strings into recipients. Need to find the sender's registry in order to get the registry defaults correct for its recipients.
    [COND
      [SENDER
        (SETQ OLDFROM (\GV.PARSERECIPIENTS SENDER NIL :BOTH))
        ; Sender is a mail address, and has the official registry
        ; Elements are of the form (prettyname gvname . registry)
        (if FROM
          then
            ; Now that we have a source of official registry (we hope), parse
            ; the From field with reference to it.
            (SETQ OLDFROM (\GV.PARSERECIPIENTS FROM (PROGN
              ; Return first element that has a registry
              (CL:SOME 'CDDR OLDFROM))
              :BOTH])
            ; Have to parse the From field before we can get its registry
            (SETQ OLDFROM (\GV.PARSERECIPIENTS FROM NIL :BOTH))
          (if (NULL OLDFROM)
            then (LAB.PROMPTPRINT MAILFOLDER T "Warning: message has no FROM field")
            else (SETQ ORIGINALREGISTRY (CL:SOME 'CDDR OLDFROM)) ; Choose first element that has a registry
              )
            (AND TO (SETQ TO (\GV.PARSERECIPIENTS TO ORIGINALREGISTRY :BOTH)))
            (AND CC (SETQ CC (\GV.PARSERECIPIENTS CC ORIGINALREGISTRY :BOTH)))
            (SETQ NEWTO (OR (AND REPLYTO (SETQ REPLYTO (\GV.PARSERECIPIENTS REPLYTO ORIGINALREGISTRY :BOTH)))
              OLDFROM))
            (SETQ NEWCC (\GV.DIFFERENCE [COND
              [REPLYTO
                (LIST (CONS (fetch (LAFITEMODEDATA FULLUSERNAME) of
                  *LAFITE-MODE-DATA*
                  )
                  (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of
                  *LAFITE-MODE-DATA*
                  ]
                (T
                  ; By default CC everyone who received the original message and
                  ; to whom we are not directly replying already
                  (APPEND TO (\GV.DIFFERENCE CC TO]
                  NEWTO))
              (LAFITE.FILL.IN.ANSWER.FORM SUBJECT (if [AND (OR (NULL REPLYTO)
                (EQUAL REPLYTO OLDFROM))
                (NULL (CDDR NEWCC))
                (OR (NULL NEWCC)
                  (LET ((CC1 (CDAR NEWCC))
                    (CC2 (fetch (LAFITEMODEDATA UNPACKEDUSERNAME)
                      of *LAFITE-MODE-DATA*))
                    (AND (STRING-EQUAL (CAR CC1)
                      (CAR CC2))
                      (STRING-EQUAL (CDDR CC1)
                      (CDDR CC2))
                    then
                      ; Replying only to sender (and maybe self), so just say "your"
                      ; instead of "Joe Bob Smith <JBSmith.pa>s"
                      NIL
                    else FROM)
                DATE NEWTO NEWCC (FUNCTION (LAMBDA (NAMES STREAM)
                  ;; Print all the pretty names
                  (LA.PRINT.COMMA.LIST (MAPCAR NAMES (FUNCTION (CAR)
                    STREAM]))
```

**(GV.DIFFERENCE**

```
[LAMBDA (A B) ; Edited 6-Jun-88 13:49 by bvm
  ;; Return all the names in A that aren't in B. Names are of the form (prettyname gvname . registry). We can't eliminate all gvname duplicates,
  ;; since sometimes the prettyname contains more information (yecch).
  (for TRIPLE in A collect TRIPLE
    unless (for OTHER in B bind (GVNAME _ (CADR TRIPLE))
      thereis (AND (STRING-EQUAL (CADR OTHER)
        GVNAME)
        (OR (PROGN
          (STRING-EQUAL (CAR OTHER)
```



```

                                (CAR TRIPLE)))
                                ; Identical gv names, and TRIPLE has no interesting additional
                                ; info, like a people name
                                (AND (STRING-EQUAL (CDDR OTHER)
                                (CDDR TRIPLE))
                                (STRING-EQUAL (CAR TRIPLE)
                                (CONCAT (CADR TRIPLE)
                                "."
                                (CDDR TRIPLE]))
                                )

)

(ADDTOVAR MAILSERVERTYPES )

(ADDTOVAR LAFITEDLDIRECTORIES )

(ADDTOVAR LAFITE.PERSONAL.VARS LAFITE.GV.FROM.FIELD)

(RPAQ? ARPANETGATEWAY.REGISTRY 'AG)

(RPAQ? LAFITEREPLYTOMENU NIL)

(RPAQ? LAFITEDL.EXT "DL")

(RPAQ? LAFITE.GV.FROM.FIELD )

(ADDTOVAR LAFITESUPPORT (GV "LafiteSupport.pa"))

(ADDTOVAR LAFITEMENUVARS LAFITEREPLYTOMENU)

(RPAQQ LAFITEREPLYTOMENUITEMS ("Send message as is" 'NO)
                                ("Reply-to: me" 'SELF "Insert a Reply-to: field instructing responder to reply
                                only to you")
                                ("Reply-to: other" 'OTHER "Edit your own Reply-to: field into the message")
                                ("Abort" 'ABORT "Don't send the message"))

(PUTPROPS MAINTAIN FILEDEF MAINTAIN)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD MAILPORT (HOST# . SOCKET#))

(RECORD GVMAILPARSE (GVPSUBJECT GVPFROM GVPFORMAT . GVPRECIPIENTS))
)

(RPAQQ MAILCLIENTCONSTANTCOMS
  ((COMS
    (CONSTANTS (\OP.ADDRECIPIENT 21)
    (\OP.ADDTOITEM 24)
    (\OP.CHECKVALIDITY 22)
    (\OP.HUMANMESS 520)
    (\OP.MSEXPNAD 27)
    (\OP.SEND 26)
    (\OP.STARTITEM 23)
    (\OP.STARTSEND 20))
    (CONSTANTS (\MAILSOCKETTIMEOUT 36000000)))
    (COMS
    (CONSTANTS (\OP.OPENINBOX 0)
    (\OP.NEXTMESSAGE 1)
    (\OP.READTOC 2)
    (\OP.READMESSAGE 3)
    (\OP.WRITETOC 4)
    (\OP.DELETMESSAGE 5)
    (\OP.FLUSH 6)))
    (COMS
    (CONSTANTS (\RC.SENDSTARTED 0)
    (\RC.PASSWORDINVALID 1)
    (\RC.SENDERNOTREGISTERED 2)
    (\RC.RETURNTO NOTREGISTERED 3)
    (\RC.COMMUNICATIONFAILURE 4)))
    (COMS
    (CONSTANTS (\RC.NAMEISGROUP 1)
    (\RC.NAMEANDPASSWORDVALID 2)
    (\RC.NAMENOTREGISTERED 3)
    (\RC.COMMUNICATIONFAILURE 4)
    (\RC.INVALIDPASSWORD 5)))
    (COMS
    (CONSTANTS (\I.POSTMARK 8)
    (\I.SENDER 16)
    (\I.RETURNTO 24)
    (\I.RECIPIENTS 32)
    (\I.TEXT 520)
    (\I.OLDTEDITFORMATTING ' (560 561))
    (\I.TEDITFORMATTING 562)
    (\I.END 65535)))
  ))
  ; Mail retrieval opcodes
  ; Mail sender opcodes
  ; return codes from 'start to send a message'
  ; return codes from 'open mail box'
  ; Message Item types

```

```
(CONSTANTS (\PT.LAURELCHECK 140)
  (\PT.NOMAILBOX 139)
  (\PT.NONEWMAIL 138)
  (\PT.NEWMAIL 137)))
```

:: Mail retrieval opcodes

```
(DECLARE%: EVAL@COMPILE

(RPAQQ \OP.ADDRECIPIENT 21)

(RPAQQ \OP.ADDTOITEM 24)

(RPAQQ \OP.CHECKVALIDITY 22)

(RPAQQ \OP.HUMANMESS 520)

(RPAQQ \OP.MSEXPAND 27)

(RPAQQ \OP.SEND 26)

(RPAQQ \OP.STARTITEM 23)

(RPAQQ \OP.STARTSEND 20)

(CONSTANTS (\OP.ADDRECIPIENT 21)
  (\OP.ADDTOITEM 24)
  (\OP.CHECKVALIDITY 22)
  (\OP.HUMANMESS 520)
  (\OP.MSEXPAND 27)
  (\OP.SEND 26)
  (\OP.STARTITEM 23)
  (\OP.STARTSEND 20))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \MAILSOCKETTIMEOUT 36000000)

(CONSTANTS (\MAILSOCKETTIMEOUT 36000000))
)
```

:: Mail sender opcodes

```
(DECLARE%: EVAL@COMPILE

(RPAQQ \OP.OPENINBOX 0)

(RPAQQ \OP.NEXTMESSAGE 1)

(RPAQQ \OP.READTOC 2)

(RPAQQ \OP.READMESSAGE 3)

(RPAQQ \OP.WRITETOC 4)

(RPAQQ \OP.DELETEMESSAGE 5)

(RPAQQ \OP.FLUSH 6)

(CONSTANTS (\OP.OPENINBOX 0)
  (\OP.NEXTMESSAGE 1)
  (\OP.READTOC 2)
  (\OP.READMESSAGE 3)
  (\OP.WRITETOC 4)
  (\OP.DELETEMESSAGE 5)
  (\OP.FLUSH 6))
)
```

:: return codes from 'start to send a message'

```
(DECLARE%: EVAL@COMPILE

(RPAQQ \RC.SENDSTARTED 0)

(RPAQQ \RC.PASSWORDINVALID 1)

(RPAQQ \RC.SENDERNOTREGISTERED 2)

(RPAQQ \RC.RETURNTONOTREGISTERED 3)

(RPAQQ \RC.COMMUNICATIONFAILURE 4)

(CONSTANTS (\RC.SENDSTARTED 0)
  (\RC.PASSWORDINVALID 1)
  (\RC.SENDERNOTREGISTERED 2))
```

```

{MEDLEY}<internal>MAILCLIENT.;1

    (\RC.RETURNNOTREGISTERED 3)
    (\RC.COMMUNICATIONFAILURE 4)
)

;; return codes from 'open mail box'

(DECLARE%: EVAL@COMPILE

(RPAQQ \RC.NAMEISGROUP 1)

(RPAQQ \RC.NAMEANDPASSWORDVALID 2)

(RPAQQ \RC.NAMENOTREGISTERED 3)

(RPAQQ \RC.COMMUNICATIONFAILURE 4)

(RPAQQ \RC.INVALIDPASSWORD 5)

(CONSTANTS (\RC.NAMEISGROUP 1)
    (\RC.NAMEANDPASSWORDVALID 2)
    (\RC.NAMENOTREGISTERED 3)
    (\RC.COMMUNICATIONFAILURE 4)
    (\RC.INVALIDPASSWORD 5))
)

;; Message Item types

(DECLARE%: EVAL@COMPILE

(RPAQQ \I.POSTMARK 8)

(RPAQQ \I.SENDER 16)

(RPAQQ \I.RETURNTO 24)

(RPAQQ \I.RECIPIENTS 32)

(RPAQQ \I.TEXT 520)

(RPAQQ \I.OLDTEDITFORMATTING (560 561))

(RPAQQ \I.TEDITFORMATTING 562)

(RPAQQ \I.END 65535)

(CONSTANTS (\I.POSTMARK 8)
    (\I.SENDER 16)
    (\I.RETURNTO 24)
    (\I.RECIPIENTS 32)
    (\I.TEXT 520)
    (\I.OLDTEDITFORMATTING '(560 561))
    (\I.TEDITFORMATTING 562)
    (\I.END 65535))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \PT.LAURELCHECK 140)

(RPAQQ \PT.NOMAILBOX 139)

(RPAQQ \PT.NONEWMAIL 138)

(RPAQQ \PT.NEWMAIL 137)

(CONSTANTS (\PT.LAURELCHECK 140)
    (\PT.NOMAILBOX 139)
    (\PT.NONEWMAIL 138)
    (\PT.NEWMAIL 137))
)

(CL:PROCLAIM '(GLOBAL MAILSERVERTYPES ARPANETGATEWAY.REGISTRY LAFITEREPLYTOMENUITEMS LAFITEREPLYTOMENU))

(CL:PROCLAIM '(CL:SPECIAL *MSGOUTSTREAM*))

(FILESLoad (SOURCE)
    LAFITEDECLS)

(FILESLoad (LOADCOMP)
    GRAPEVINE PUP BSP)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)

```

{MEDLEY}<internal>MAILCLIENT.;1

Page 20

(DECLARE%: DONTEVAL@LOAD DOCOPY

(CL:PROCLAIM ' (GLOBAL LAFITE.GV.FROM.FIELD LAFITEDL.EXT LAFITEDLDIRECTORIES))

(CL:PROCLAIM ' (CL:SPECIAL \*GV-SHOW-POSTMARK\*))

(FILESLOAD GRAPEVINE)

)

(PUTPROPS **MAILCLIENT COPYRIGHT** ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1989 1990))

---

### FUNCTION INDEX

|                          |    |                               |    |                                  |    |
|--------------------------|----|-------------------------------|----|----------------------------------|----|
| GETMAILSERVEROPS .....   | 8  | GV.STARTSEND .....            | 1  | \GV.PARSE.SIMPLE.RECIPIENT ..... | 13 |
| GV.ADDRECIPIENT .....    | 2  | GV.WRITETOC .....             | 5  | \GV.PASERECIPIENTS .....         | 11 |
| GV.ADDTOITEM .....       | 2  | MS.EXPAND .....               | 2  | \GV.PASERECIPIENTS1 .....        | 11 |
| GV.CHECKVALIDITY .....   | 2  | MS.RETRIEVEOPERATION .....    | 5  | \GV.SEND.PARSE .....             | 10 |
| GV.CLOSEMAILBOX .....    | 5  | MS.SENDOPERATION .....        | 3  | \GV.SENDMESSAGE .....            | 8  |
| GV.DELETMESSAGE .....    | 5  | \CACHED.HOST.NAME .....       | 7  | \GV.SENDRECIPIENTS .....         | 10 |
| GV.INIT.MAIL.USER .....  | 7  | \CONNECTOMAILSERVER .....     | 5  | \LAFITE.CHOOSEREPLYTO .....      | 15 |
| GV.MAKEANSWERFORM .....  | 16 | \FINDMAILSERVER .....         | 3  | \MAILSERVERSOCKETS .....         | 3  |
| GV.NEXTMESSAGE .....     | 5  | \GV.DIFFERENCE .....          | 16 | \RECEIVEACK .....                | 3  |
| GV.OPENMAILBOX .....     | 4  | \GV.EXTRACT.FIELD .....       | 14 | \RECEIVELONGWORD .....           | 6  |
| GV.POLLNEWMAIL .....     | 4  | \GV.FIND.NON.SPACE .....      | 13 | \RESPTOCHECKVAL .....            | 3  |
| GV.PORTFROMNAME .....    | 4  | \GV.HANDLE.DL .....           | 14 | \RESPTOEXPAND .....              | 3  |
| GV.READTOC .....         | 5  | \GV.MAILSERVERTYPE .....      | 8  | \RESPTONEXTMESSAGE .....         | 6  |
| GV.RETRIEVEMESSAGE ..... | 5  | \GV.MESSAGE.FROM.SELF.P ..... | 15 | \RESPTOOPENMAILBOX .....         | 6  |
| GV.SEND .....            | 2  | \GV.MESSAGE.P .....           | 15 | \RESPTORETRIEVEMESSAGE .....     | 6  |
| GV.STARTITEM .....       | 2  | \GV.PARSE.FAILED .....        | 15 | \RESPTOSTARTSEND .....           | 3  |

---

### CONSTANT INDEX

|                             |    |                       |    |                                 |       |
|-----------------------------|----|-----------------------|----|---------------------------------|-------|
| \I.END .....                | 19 | \OP.FLUSH .....       | 18 | \PT.NOMAILBOX .....             | 19    |
| \I.OLDTEDITFORMATTING ..... | 19 | \OP.HUMANMESS .....   | 18 | \PT.NONEWMAIL .....             | 19    |
| \I.POSTMARK .....           | 19 | \OP.MSEXPAEND .....   | 18 | \RC.COMMUNICATIONFAILURE .....  | 18,19 |
| \I.RECIPIENTS .....         | 19 | \OP.NEXTMESSAGE ..... | 18 | \RC.INVALIDPASSWORD .....       | 19    |
| \I.RETURNTO .....           | 19 | \OP.OPENINBOX .....   | 18 | \RC.NAMEANDPASSWORDVALID .....  | 19    |
| \I.SENDER .....             | 19 | \OP.READMESSAGE ..... | 18 | \RC.NAMEISGROUP .....           | 19    |
| \I.TEDITFORMATTING .....    | 19 | \OP.READTOC .....     | 18 | \RC.NAMENOTREGISTERED .....     | 19    |
| \I.TEXT .....               | 19 | \OP.SEND .....        | 18 | \RC.PASSWORDINVALID .....       | 18    |
| \MAILSOCKETTIMEOUT .....    | 18 | \OP.STARTITEM .....   | 18 | \RC.RETURNTONOTREGISTERED ..... | 18    |
| \OP.ADDRECIPIENT .....      | 18 | \OP.STARTSEND .....   | 18 | \RC.SENDERNOTREGISTERED .....   | 18    |
| \OP.ADDTOITEM .....         | 18 | \OP.WRITETOC .....    | 18 | \RC.SENDSTARTED .....           | 18    |
| \OP.CHECKVALIDITY .....     | 18 | \PT.LAURELCHECK ..... | 19 |                                 |       |
| \OP.DELETMESSAGE .....      | 18 | \PT.NEWMAIL .....     | 19 |                                 |       |

---

### VARIABLE INDEX

|                               |    |                              |    |                              |      |                               |     |
|-------------------------------|----|------------------------------|----|------------------------------|------|-------------------------------|-----|
| *GV-SHOW-POSTMARK* .....      | 7  | LAFITEDLDIRECTORIES .....    | 17 | MAILCLIENTCONSTANTCOMS ..... | 17   | \MAILSERVERPOLLINGSOC .....   | 4   |
| ARPANETGATEWAY.REGISTRY ..... | 17 | LAFITEMENUVARS .....         | 17 | MAILSERVERTYPES .....        | 5,17 | \MAILSERVERRETRIEVALSOC ..... | 4   |
| GV.MAILBOX.TIMEOUT .....      | 7  | LAFITEMODELST .....          | 7  | \CACHED.HOST.NAMES .....     | 7    | \MAILSERVERSOCKETCACHE .....  | 4   |
| LAFITE.GV.FROM.FIELD .....    | 17 | LAFITEREPLYTOMENU .....      | 17 | \MAILIOTIMEOUT .....         | 4    | \SYSTEMCACHEVARS .....        | 4,7 |
| LAFITE.PERSONAL.VARS .....    | 17 | LAFITEREPLYTOMENUITEMS ..... | 17 | \MAILSERVERENQUIRYSOC .....  | 4    |                               |     |
| LAFITEDL.EXT .....            | 17 | LAFITESUPPORT .....          | 17 | \MAILSERVERNAME .....        | 4    |                               |     |

---

### RECORD INDEX

|                   |    |                |    |
|-------------------|----|----------------|----|
| GVMAILPARSE ..... | 17 | MAILPORT ..... | 17 |
|-------------------|----|----------------|----|

---

### PROPERTY INDEX

|                |    |
|----------------|----|
| MAINTAIN ..... | 17 |
|----------------|----|

---