

File created: 23-Feb-2024 22:24:01 {WMEDLEY}<library>lafite>LAFITE-MAIL.;2

edit by: rmk

changes to: (VARS LAFITE-MAILCOMS)

previous date: 23-Feb-2024 22:05:03 {WMEDLEY}<library>lafite>LAFITE-MAIL.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **LAFITE-MAILCOMS**

```
((COMS ; Retrieving mail
(FNS \LAFITE.GETMAIL \LAFITE.GETMAIL.FROM.ICON \LAFITE.GETMAIL.PROC \LAFITE.GETNEWMAIL
 \LAFITE.GETNEWMAIL1 \LAFITE.GETNEWMAIL# \LAFITE.RETRIEVEMESSAGES))
(COMS ; Mail polling and registration
(FNS \LAFITE.GET.USER.DATA \LAFITE.GUESS.MODE \LAFITE.REGISTER.MODE LAFITECLEARCACHE FULLUSERNAME
LAFITE.USER.NAME.FROM.LOGIN LAFITEMAILWATCH \LAFITE.WAKE.WATCHER POLLNEWMAIL
 \LAFITE.NEW.MAIL.EXISTS PRINTLAFITESTATUS LAFITE.STATUS.WITH.TIME \LAFITE.REINITIALIZING))
[COMS ; Parsing mail files
(FNS \LAFITE.PARSE.FOLDER \LAFITE.PARSE.FOLDER1 \LAFITE.HANDLE.DUPLICATES \LAFITE.CHECK.DUPLICATE
 \LAFITE.REPORT.DUPLICATES BADMAILFILE BADMAILFILE.CLOSEFN BADMAILFILE.FLAGBYTE
VERIFYMAILFOLDER VERIFYFAILED \LAFITE.READ.TOC.FILE BADTOCFILE \LAFITE.TOC.EOF.LA.READCOUNT
LA.READSTAMP LA.PRINTHEADER LA.PRINTCOUNT LA.PRINTSTAMP LA.READSHORTSTRING
LA.PRINTSHORTSTRING.LA.READSTRING \LAFITE.VERIFYMSG.LA.MSGFROMMEP)
(FNS LAFITE.PARSE.MSG.FOR.TOC LAFITE-EXTRACT-REAL-NAME LAFITE.FETCH.TO.FIELD LAFITE.PARSE.HEADER
LAFITE.GRAB.DATE LAFITE.READ.LINE.FOR.TOC LAFITE.READ.FORMAT LAFITE.READ.NAME.FIELD
LAFITE.READ.ONE.LINE.FOR.TOC LAFITE.READ.TO.EOL.LA.SKIP.TO.EOL LAFITE.SKIP.WHITE.SPACE)
(FNS \LAFITE.PARSE.MESSAGE)
(COMS (VARS LA.FULLPARSEFIELDS LA.TOCFIELDS LA.TOFIELDONLY LA.DATEFIELDONLY LA.SUBJECTFIELDONLY)
(FNS LAFITE.INIT.PARSETABLES LAFITE.MAKE.PARSE.TABLE LAFITE.MAKE.PARSE.TABLE1))
(COMS ; New header parser
(FNS LAFITE.NEW.PARSE.HEADER LAFITE.HANDLE.ORIGINAL.FIELD)
(INITVARS (*LAFITE-MAX-FIELD-WIDTH* 100)
(*LAFITE-PARSE-HEADER-STRING-RESOURCE*))
(COMS (INITVARS (*LAFITE-VERIFY-ACTION* NIL)
(MAILWATCHWAITTIME 5)
(LAFITEFLUSHMAILFLG T)
(LAFITETOC.EXT "-Lafite-toc")
(LAFITENEWMAILFN NIL)
(LAFITENEWMAILTUNE NIL)
(LAFITEGETMAILTUNE NIL)
(LAFITE.AFTER.GETMAIL.FN NIL)
(LAFITE.SORT.NEW.MAIL NIL))
(INITVARS (\LAFITE.LAST.STATUS))
(ADDVARS (\SYSTEMCACHEVARS \LAFITE.LAST.STATUS)))
[DECLARE%: DOEVAL@COMPILE (P (CL:PROCLAIM ' (CL:SPECIAL *LAFITE-VERIFY-ACTION* DEFAULTREGISTRY
LAFITEDEBUGFLG))
(CL:PROCLAIM ' (GLOBAL LAFITEFLUSHMAILFLG LAFITEGETMAILTUNE LAFITENEWMAILFN
LAFITENEWMAILTUNE LAFITEIFFFROMMETHENSEENFLG
MAILWATCHWAITTIME LAFITETOC.EXT LAFITE.SORT.NEW.MAIL]
(DECLARE%: EVAL@COMPILE DONTCOPY
(GLOBALVARS *LAFITE-MAX-FIELD-WIDTH* *LAFITE-PARSE-HEADER-STRING-RESOURCE* LA.FULLPARSEFIELDS
LA.TOCFIELDS LA.TOFIELDONLY LA.DATEFIELDONLY LA.SUBJECTFIELDONLY
\LAFITE.AUTHENTICATION.FAILURE \LAPARSE.FULL \LAPARSE.TOCFIELDS \LAPARSE.TOFIELD
\LAPARSE.SUBJECTFIELD \LAPARSE.DATEFIELD LAFITE.AFTER.GETMAIL.FN)
(FILES (SOURCE)
LAFITE-DECLS)
(LOCALVARS . T))))
```

:: Retrieving mail

(DEFINEQ

(\LAFITE.GETMAIL

```
[LAMBDA (WINDOW MAILFILEDATA ITEM MENU) ; (* bvm%: "25-Mar-84 17:20")
(\LAFITE.PROCESS (LIST (FUNCTION \LAFITE.GETMAIL.PROC)
(KWOTE WINDOW)
(KWOTE MAILFILEDATA)
(KWOTE ITEM)
(KWOTE MENU))
'LAFITEGETMAIL])
```

(\LAFITE.GETMAIL.FROM.ICON

```
[LAMBDA (ICONW) ; Edited 3-Jun-88 12:16 by bvm
;; Called from icon menu--expand the window and run GetMail.
(LAB.DO.COMMAND (PROG1 (WINDOWPROP ICONW 'ICONFOR)
(EXPANDW ICONW))
(FUNCTION \LAFITE.GETMAIL])
```

(\LAFITE.GETMAIL.PROC

```
[LAMBDA (WINDOW MAILFOLDER ITEM MENU) ; (* bvm%: "11-Nov-84 18:30")
  (RESETLST
   (LA.RESETSHADE ITEM MENU)
   (OBTAIN.MONITORLOCK (fetch FOLDERLOCK of MAILFOLDER)
    NIL T)
   (\LAFITE.MAYBE.CLEAR.PROMPT MAILFOLDER)
   (OBTAIN.MONITORLOCK \LAFITE.MAILSERVERLOCK NIL T)
   (\LAFITE.GETNEWMAIL MAILFOLDER WINDOW))
 (\LAFITE.WAKE.WATCHER)]
```

(\LAFITE.GETNEWMAIL

; Edited 1-May-89 11:36 by bvm

```
[LAMBDA (FOLDER WINDOW)
  (PROG* ((ALLMODES (LAFITE.ALL.MODES.P :GETMAIL))
   [NEWMAILSEEN (for MODE in \LAFITE.ACTIVE.MODES when (OR ALLMODES (EQ (fetch (LAFITEMODEDATA LAFITEOPS)
    of MODE)
    \LAFITEMODE))
   thereis (for MAILSERVER in (fetch (LAFITEMODEDATA MAILSERVERS) of MODE)
    thereis (fetch (MAILSERVER NEWMAILP) of MAILSERVER]
   (NBOXES 0)
   (FIRST# (ADD1 (fetch (MAILFOLDER %OFMESSAGES) of FOLDER)))
   ALLMESSAGELIST FIRSTMESSAGE SORTED)
   (\LAFITE.OPEN.FOLDER FOLDER 'APPEND :OK)
   (replace (MAILFOLDER FOLDERGETSMail) of FOLDER with T)
   [for *LAFITE-MODE-DATA* in \LAFITE.ACTIVE.MODES when (OR ALLMODES (EQ (fetch (LAFITEMODEDATA LAFITEOPS)
    of *LAFITE-MODE-DATA*)
    \LAFITEMODE))
   do (for MAILSERVER in (fetch (LAFITEMODEDATA MAILSERVERS) of *LAFITE-MODE-DATA*) bind MESSAGELIST
    when [AND (OR (NOT NEWMAILSEEN)
    (fetch (MAILSERVER NEWMAILP) of MAILSERVER))
    (PROGN ; i.e., only here if NOBODY reported mail (in which case user is
    ; asking for explicit poll), or if watcher already noticed mail
    (if (> NBOXES 0)
     then (LAB.PROMPTPRINT FOLDER "; ")
     (add NBOXES 1)
     (LAB.PROMPTPRINT FOLDER (fetch (MAILSERVER MAILSERVERNAME) of MAILSERVER)
      ". ")]
    (SETQ MESSAGELIST (\LAFITE.GETNEWMAIL1 MAILSERVER FOLDER FIRSTMESSAGE]
   do (LAB.APPENDMESSAGES FOLDER MESSAGELIST)
    (SETQ ALLMESSAGELIST (NCONC ALLMESSAGELIST MESSAGELIST]
   (if (EQ NBOXES 0)
    then ; No mode had any mail servers
    (LAB.PROMPTPRINT FOLDER "No mailboxes known")
   elseif ALLMESSAGELIST
    then (if (AND (CDR ALLMESSAGELIST)
    (SELECTQ LAFITE.SORT.NEW.MAIL
    (NIL NIL)
    (:MULTIPLE (> NBOXES 1))
    T))
    then ; Sort the newly arrived messages by date
    (LAB.PROMPTPRINT FOLDER "; ")
    (LAFITE.SORT.BY.DATE FOLDER FIRST#)
    (SETQ SORTED T))
    (LET ((MESSAGES (fetch (MAILFOLDER MESSAGEDESCRIPTORS) of FOLDER)))
     (if LAFITE.AFTER.GETMAIL.FN
      then ; User hook for mail filtering
      (CL:FUNCALL LAFITE.AFTER.GETMAIL.FN FOLDER
       (if SORTED
        then ; Recompute this list in the new order
        (for I from FIRST# to (fetch (MAILFOLDER %OFMESSAGES)
         of FOLDER)
         collect (NTHMESSAGE MESSAGES I))
        else ALLMESSAGELIST)))
     (if (< (fetch (MAILFOLDER LASTSELECTEDMESSAGE) of FOLDER)
      (SETQ FIRSTMESSAGE (fetch (MAILFOLDER FIRSTSELECTEDMESSAGE) of FOLDER)))
      then ; Select the first message and make sure it is visible
      (SELECTMESSAGE (SETQ FIRSTMESSAGE (NTHMESSAGE MESSAGES FIRST#))
       FOLDER)
     else ; Hook must have selected something
     (SETQ FIRSTMESSAGE (NTHMESSAGE MESSAGES FIRSTMESSAGE)))
    (LAB.EXPOSEMESSAGE FOLDER FIRSTMESSAGE)
    (COND
     (LAFITEGETMAILTUNE (PLAYTUNE LAFITEGETMAILTUNE)))
    (PRINTLAFITESTATUS "Finished Retrieving Mail")))
   (LAB.PROMPTPRINT FOLDER ". "])]
```

(\LAFITE.GETNEWMAIL1

; Edited 24-Oct-88 17:43 by bvm

```
[LAMBDA (MAILSERVER MAILFOLDER NTHTIME)
  (PROG (MESSAGELIST OPENRESULT MAILBOX %OFMESSAGES OUTSTREAM)
   (SETQ OPENRESULT (CL:FUNCALL (fetch (MAILSERVER OPENMAILBOX) of MAILSERVER)
    (fetch (MAILSERVER MAILPORT) of MAILSERVER)
    (fetch (LAFITEMODEDATA FULLUSERNAME) of *LAFITE-MODE-DATA*)
    (fetch (LAFITEMODEDATA CREDENTIALS) of *LAFITE-MODE-DATA*)
    MAILSERVER))
  [SELECTQ (COND
   ((LISTP OPENRESULT)
```

```

      (SETQ MAILBOX (fetch (OPENEDMAILBOX MAILBOX) of OPENRESULT)))
      (T OPENRESULT))
(EMPTY ; Nothing to retrieve
(LAB.PROMPTPRINT MAILFOLDER "empty")
(RETURN NIL))
(NIL ; No response
)
(COND
  (MAILBOX (COND
    ((NOT NTHTIME)
      (PRINTLAFITESTATUS "Retrieving Mail")
      (UNSELECTALLMESSAGES MAILFOLDER)))
    (\LAFITE.OPEN.FOLDER MAILFOLDER 'APPEND :OK)
    (COND
      ((SETQ %OFMESSAGES (LISTGET (fetch (OPENEDMAILBOX PROPERTIES) of OPENRESULT)
        '%OFMESSAGES))
        (\LAFITE.GETNEWMAIL# MAILFOLDER %OFMESSAGES)))
      (RETURN (COND
        ((SETQ MESSAGELIST (\LAFITE.RETRIEVEMESSAGES MAILSERVER MAILBOX MAILFOLDER))
          [COND
            ((NULL %OFMESSAGES)
              (\LAFITE.GETNEWMAIL# MAILFOLDER (LENGTH MESSAGELIST)
                MESSAGELIST)
              (LAB.PROMPTPRINT MAILFOLDER "not responding"))
            (COND
              ((CDR (LISTP OPENRESULT)) ; Say more about why not responding
                (LAB.PROMPTPRINT MAILFOLDER " (" (fetch (OPENEDMAILBOX PROPERTIES) of OPENRESULT)
                  ") "]))
          ]))
        ]))

```

(\LAFITE.GETNEWMAIL#

```

[LAMBDA (MAILFOLDER %OFMESSAGES) ; (* bvm%: " 4-Feb-86 12:17")
  (LAB.PROMPTPRINT MAILFOLDER "(" %OFMESSAGES (COND
    ((EQ %OFMESSAGES 1)
      " msg")
    (T " msgs"))
    ") ")]

```

(\LAFITE.RETRIEVEMESSAGES

```

[LAMBDA (MAILSERVER MAILBOX FOLDER) ; Edited 28-Apr-89 11:54 by bvm
  (LET*
    ((OUTSTREAM (\LAFITE.OPEN.FOLDER FOLDER 'APPEND :OK))
      (ORIGEOF (GETEOFPTR OUTSTREAM))
      SUCCESS)
    [if (NOT (= ORIGEOF (fetch (MAILFOLDER FOLDEREOFPTR) of FOLDER)))
      then ; Oops, something snuck in here. Ordinarily this is caught when
          ; we open the file, so probably this is result of internal bug
      (SETQ ORIGEOF (GETEOFPTR (SETQ OUTSTREAM (\LAFITE.FOLDER.CHANGED FOLDER OUTSTREAM 'BOTH :OK)
        (CL:UNWIND-PROTECT
          [CATCH-ABORT "Abort retrieval from this mailbox"
            (LET ((WINDOW (fetch (MAILFOLDER BROWSERPROMPTWINDOW) of FOLDER))
              (NEXTMESSAGEFN (fetch (MAILSERVER NEXTMESSAGE) of MAILSERVER))
              (RETRIEVEFN (fetch (MAILSERVER RETRIEVEMESSAGE) of MAILSERVER))
              (ENDPOS ORIGEOF)
              (COUNTER 0)
              (MODEBITS (fetch (LAFITEMODEDATA MODEINDEX) of *LAFITE-MODE-DATA*)))
              XPOS MESSAGELIST STARTPOS LENGTHPOS MSGLENGTH NEXTMESSAGERESULT MSG)
            [if WINDOW
              then ; We're going to print a number here each time we retrieve a
                  ; message to show how far we've gotten.
              (SETQ XPOS (DSPXPOSITION NIL WINDOW)) ; Number will start here
              (if [< (- (DSPRIGHTMARGIN NIL WINDOW)
                XPOS)
                (TIMES (CHARWIDTH (CHARCODE 0)
                  WINDOW)
                  (LET [(N (LISTGET (fetch (OPENEDMAILBOX PROPERTIES) of MAILBOX)
                    '%OFMESSAGES))
                    (if N
                      then (NCHARS N)
                      else 2]
                    then
                      ;; we're about to run off the right edge of the prompt window, so go to a new line. Guess about width
                      ;; is according to how large a number we expect ultimately to print.
                      (LAB.PROMPTPRINT FOLDER "
                        ")
                      (SETQ XPOS (DSPXPOSITION NIL WINDOW))
                    (while (SETQ NEXTMESSAGERESULT (CL:FUNCALL NEXTMESSAGEFN MAILBOX))
                      unless (AND (LISTP NEXTMESSAGERESULT)
                        (LISTGET NEXTMESSAGERESULT 'DELETED))
                        do ; print the message stamp to the file
                          (SETFILEPTR OUTSTREAM (SETQ STARTPOS ENDPOS))
                          [COND
                            ((NOT (= STARTPOS (GETEOFPTR OUTSTREAM)))
                              (HELP "Lafite is confused about where the end of the file is."))
                            ]))

```

```

; If the user cleverly returns from here, god help us
      (SETFILEPTR OUTSTREAM (SETQ STARTPOS (GETEOFPTR OUTSTREAM))
      (LA.PRINTSTAMP OUTSTREAM)
      (SETQ LENGTHPOS (GETFILEPTR OUTSTREAM))
      (PRIN3 "00000027 00027 UU " OUTSTREAM)
      (BOUT OUTSTREAM (CHARCODE CR)) ; now get the message and put it in the file
      (CL:FUNCALL RETRIEVEFN MAILBOX OUTSTREAM)
      (SETQ MSGLENGTH (- (SETQ ENDPOS (GETFILEPTR OUTSTREAM))
      STARTPOS)) ; go back and print the message length in the stamp
      (SETQ MSG (create LAFITEMSG
      MARKCHAR _ UNSEENMARK
      BEGIN _ STARTPOS
      STAMPLENGTH _ (+ LAFITEBASICSTAMPLENGTH 8)
      MESSAGELENGTH _ MSGLENGTH
      MODEBITS _ MODEBITS))
      (COND
      (> MSGLENGTH 99999999) ; Too big for this format to handle. You're probably in trouble
      ; already with a 100MB message! The file will be unparseable.
      ; We should probably issue some sort of warning
      (LAB.FORMAT FOLDER "~%%Warning! Lafite has just retrieved a message ~D bytes
      long, which means this folder is now in an inconsistent format. You must
      Delete the message~@[ or Move it into another folder~] before
      Updating.~%%" MSGLENGTH (TYPENAMEP MSGLENGTH 'FIXP))
      (SETQ MSGLENGTH 99999999))
      (SETFILEPTR OUTSTREAM LENGTHPOS)
      (LA.PRINTCOUNT MSGLENGTH OUTSTREAM '(FIX 8 10 T))
      (push MESSAGELIST MSG)
      [COND
      (XPOS (DSPXPOSITION XPOS WINDOW)
      (printout WINDOW .i1 (add COUNTER 1])
      finally (COND
      (XPOS ; Prepare to overwrite counter with 'done'
      (DSPXPOSITION XPOS WINDOW)))
      (SETQ SUCCESS T)
      (RETURN (REVERSE MESSAGELIST)

```

:: Cleanups: Do this whether we were successful or not

```

      (if (NULL SUCCESS)
      then ; Retrieval error somewhere. Dispose of what we have retrieved
      (LAB.PROMPTPRINT FOLDER " retrieval aborted")
      (SETFILEPTR OUTSTREAM ORIGEOF)
      (SETFILEINFO OUTSTREAM 'LENGTH ORIGEOF))
      (\LAFITE.CLOSE.FOLDER FOLDER (NULL SUCCESS)) ; Force output on the mail file so we're sure we have it
      [LET* [(FLUSHP (AND SUCCESS LAFITEFLUSHMAILFLG))
      (FLUSHED (IGNORE-ERRORS
      (CL:FUNCALL (fetch (MAILSERVER CLOSEMAILBOX) of MAILSERVER)
      MAILBOX FLUSHP)]
      (if SUCCESS
      then (LAB.FORMAT FOLDER "done~@[, mailbox maybe not flushed~]" (AND FLUSHP (NOT FLUSHED))))

```

:: Mail polling and registration

(DEFINEQ

(\LAFITE.GET.USER.DATA

[LAMBDA (MODE DONTWAIT RECOMPUTE)

; Edited 26-May-92 12:21 by bvm

::: Return the mode data for specified MODE, or the current mode if NIL. This function is in charge of setting \LAFITEUSERDATA

```

      (COND
      ([NOT (OR (LISTP MODE)
      (SETQ MODE (if MODE
      then (ASSOC MODE LAFITEMODELST)
      else (OR \LAFITEMODE (\LAFITE.INFER.MODE))
      (SETQ \LAFITE.AUTHENTICATION.FAILURE "No Mode")
      NIL)
      ((AND (NOT RECOMPUTE)
      (ASSOC MODE \LAFITE.ACTIVE.MODES)))
      ((NOT DONTWAIT)
      (WITH.MONITOR \LAFITE.MAILSERVERLOCK ; Let's not have everyone try this at once
      (ALLOW.BUTTON.EVENTS) ; Make sure not to trap the mouse
      [LET ((DATA (ASSOC MODE \LAFITE.ACTIVE.MODES)))
      (if RECOMPUTE
      then (SETQ \LAFITE.ACTIVE.MODES (DREMOVE DATA \LAFITE.ACTIVE.MODES))
      (SETQ DATA NIL))
      (OR DATA (LET [(HOW (CONS "Authenticating" (AND (OR (NEQ MODE \LAFITEMODE)
      (LAFITE.SHOW.MODE.P))
      (LIST " " (fetch (LAFITEOPS LAFITEMODE)
      of MODE))
      (\LAFITE.REGISTER.MODE MODE) ; In case it hasn't been done yet
      (PRINTLAFITESTATUS (LAFITE.STATUS.WITH.TIME (CONCATLIST HOW)))
      (if (SETQ DATA (CL:FUNCALL (fetch (LAFITEOPS AUTHENTICATOR) of MODE)))
      then (replace (LAFITEMODEDATA LAFITEOPS) of DATA with MODE)
      (push \LAFITE.ACTIVE.MODES DATA)
      [PRINTLAFITESTATUS (LAFITE.STATUS.WITH.TIME (CONCATLIST (RPLACA HOW

```

ated"]

(\LAFITE.WAKE.WATCHER) ; To update status window  
DATA]])

(\LAFITE.GUESS.MODE

[LAMBDA (MSG) ; Edited 9-May-88 18:40 by bvm

:: Try to figure out the mode of the message. If we're sure about it, fix the message, too.

(if (NULL (CDR \LAFITE.ACTIVE.MODES))  
then ; Only one mode, assume it's that one, but don't bother recording  
; this fact  
(fetch (LAFITEMODEDATA LAFITEMODE) of (CAR \LAFITE.ACTIVE.MODES))  
else (LET (BESTMODE OKMODE)  
[for \*LAFITE-MODE-DATA\* in \LAFITE.ACTIVE.MODES  
do (CASE (CL:FUNCALL (fetch (LAFITEMODEDATA MESSAGEP) of \*LAFITE-MODE-DATA\*))  
MSG  
(T) ; Definitely this type  
(RETURN (SETQ BESTMODE \*LAFITE-MODE-DATA\*))  
(? ; Could be this type  
(if (NOT OKMODE)  
then (SETQ OKMODE \*LAFITE-MODE-DATA\*)  
else (SETQ OKMODE T)))]  
(if (OR BESTMODE (AND (SETQ BESTMODE OKMODE)  
(NEQ BESTMODE T)))  
then ; Found it, or found an ok one with no competitors.  
(replace (LAFITEMSG MODEBITS) of MSG with (fetch (LAFITEMODEDATA MODEINDEX) of BESTMODE))  
(fetch (LAFITEMODEDATA LAFITEMODE) of BESTMODE]))

(\LAFITE.REGISTER.MODE

[LAMBDA (MODEDATA) ; Edited 6-May-88 15:15 by bvm

:: Take note of this element of LAFITEMODELST. Currently this just means adding the mode to the index-to-name list  
::\*LAFITE-WELL-KNOWN-MODES\*.

(if (NOT (FMEMB (fetch (LAFITEOPS LAFITEMODE) of MODEDATA)  
\*LAFITE-WELL-KNOWN-MODES\*))  
then ; Register this mode  
(LET ((N (fetch (LAFITEOPS MODEINDEX) of MODEDATA)))  
(while (<= (LENGTH \*LAFITE-WELL-KNOWN-MODES\*)  
N)  
do ; Make sure mode list has at least n+1 elements (zeroth elt is NIL  
; for mode = unknown).  
(SETQ \*LAFITE-WELL-KNOWN-MODES\* (NCONC1 \*LAFITE-WELL-KNOWN-MODES\* NIL)))  
(CL:SETF (CL:NTH N \*LAFITE-WELL-KNOWN-MODES\*)  
(fetch (LAFITEOPS LAFITEMODE) of MODEDATA]  
MODEDATA]))

(\LAFITE.CLEARCACHE

[LAMBDA (RECURSIVEP) ; Edited 13-Jun-88 12:47 by bvm

:: Called when login has changed, or we otherwise want to reauthenticate. If WAKEFLG, then recompute them right now.

(RESETLST  
[if (OBTAIN.MONITORLOCK \LAFITE.MAILSERVERLOCK (NULL RECURSIVEP)  
T)  
then (SETQ \LAFITE.ACTIVE.MODES NIL)  
(\LAFITE.WAKE.WATCHER)  
else ; Spawn process to do it when the lock becomes free  
(\LAFITE.PROCESS `((FUNCTION LAFITE.CLEARCACHE)  
T]))

(\FULLUSERNAME

[LAMBDA (UNPACKEDFLG MODE) ; Edited 3-May-89 18:36 by bvm

[if (NOT MODE)  
then (SETQ MODE \LAFITEMODE)  
elseif (LITATOM MODE)  
then (SETQ MODE (OR (ASSOC MODE LAFITEMODELST)  
(\ILLEGAL.ARG MODE])  
(LET (DATA)  
(COND  
[ (AND \LAFITE.ACTIVE.MODES (SETQ DATA (ASSOC MODE \LAFITE.ACTIVE.MODES)) )  
(COND  
(UNPACKEDFLG (fetch (LAFITEMODEDATA UNPACKEDUSERNAME) of DATA))  
(T (fetch (LAFITEMODEDATA FULLUSERNAME) of DATA]  
( (AND (OR MODE (SETQ MODE (\LAFITE.INFER.MODE)))  
(EQ (fetch (LAFITEOPS LAFITEMODE) of MODE)  
'NS)) ; Special-case NS username when not yet authenticated--use  
; login name with colon, letting sendmail do the defaulting later  
(CONCAT (CAR (LAFITE.USER.NAME.FROM.LOGIN T))  
":"))  
(T (LAFITE.USER.NAME.FROM.LOGIN UNPACKEDFLG]))

(\LAFITE.USER.NAME.FROM.LOGIN

```
[LAMBDA (UNPACKEDFLG RECOMPUTE) ; Edited 7-Sep-88 11:49 by bvm
;; Return name of current logged in user. If UNPACKEDFLG return it as a gv cons, else a string fullname.reg
[if (OR RECOMPUTE (NULL \LAFITE.CURRENT.USER))
  then (\INTERNAL/GETPASSWORD) ; Insure logged in
      (LET ((USER (USERNAME NIL NIL T))
            (DOT REGISTRY SIMPLENAME))
          [COND
            ((NOT (SETQ DOT (STRPOS "." USER)))
             (SETQ SIMPLENAME USER)
             (SETQ REGISTRY DEFAULTREGISTRY))
            (T (SETQ SIMPLENAME (SUBSTRING USER 1 (SUB1 DOT)))
               (SETQ REGISTRY (SUBATOM USER (ADD1 DOT)))
              )
           ]
         [COND
            ((U-CASEP SIMPLENAME) ; If user had the caps lock on when logging in, lowercase the
             ; name
              (SETQ SIMPLENAME (L-CASE SIMPLENAME T))
            ]
          [if REGISTRY
            then ; Silly grapevine code requires registry be a symbol. Make it
              ; lowercase to be nice
                (SETQ REGISTRY (MKATOM (L-CASE REGISTRY)))
              (SETQ \LAFITE.CURRENT.USER (CONS (if REGISTRY
                                                then (CONCAT SIMPLENAME "." REGISTRY)
                                                else SIMPLENAME)
                                              (CONS SIMPLENAME REGISTRY)))
            ]
          ]
        (COND
          (UNPACKEDFLG (CDR \LAFITE.CURRENT.USER))
          (T (CAR \LAFITE.CURRENT.USER)))
        ]
      ]
    ]
  ]

```

(LAFITEMAILWATCH

```
[LAMBDA NIL ; Edited 13-Jun-88 11:05 by bvm
[bind (INTERVAL _ (ITIMES MAILWATCHWAITTIME 60000))
      (FIRSTTIME _ T)
      CONTINUANCE while (PROGN ; Until killed
                          T)
do (SETQ CONTINUANCE (WITH.MONITOR \LAFITE.MAILSERVERLOCK (POLLNEWMAIL FIRSTTIME)))
   (BLOCK (if (AND CONTINUANCE (< CONTINUANCE INTERVAL))
              then ; Some server wants to be contacted within this period
                CONTINUANCE
              else INTERVAL))
   (SETQ FIRSTTIME NIL)]

```

(\LAFITE.WAKE.WATCHER

```
[LAMBDA NIL ; Edited 13-Jun-88 12:41 by bvm
;; Wakes the LAFITEMAILWATCH process in response to various actions
(PROG [(P (FIND.PROCESS 'LAFITEMAILWATCH)
          (COND
            (P (WAKE.PROCESS P))
            ((EQ \LAFITE.ACTIVE T) ; Process got killed somehow; reinstate it
             (\LAFITE.PROCESS (LIST (FUNCTION LAFITEMAILWATCH))
                                  NIL T 'HARDRESET))
          ]
      ]

```

(POLLNEWMAIL

```
[LAMBDA (RESTARTFLG) ; Edited 16-Aug-89 17:03 by bvm
;; Poll for new mail. Value returned, if non-NIL, is the "continuance"--the number of milliseconds within which some server would like to be
;; contacted again.
(PROG ((ALLMODES (LAFITE.ALL.MODES.P :POLL))
       PRIMARYMODE FAILEDMODES NOTUPFLG NOMAILFLG NEWMAILMODES MINCONTINUANCE STATUS)
      (if (OR RESTARTFLG (NULL \LAFITE.ACTIVE.MODES))
          then ; Need to get authenticated
            (\LAFITE.GET.USER.DATA)
            [if ALLMODES
              then ; Also make sure to get data for non-primary modes
                (SETQ FAILEDMODES (for MODE in LAFITEMODELST unless (OR (NLISTP (CDR MODE))
                                                                           (PROGN (SETQ
                                                                           \LAFITE.AUTHENTICATION.FAILURE
                                                                           NIL)
                                                                           (\LAFITE.GET.USER.DATA
                                                                           MODE))))
              ]
            collect (CONS \LAFITE.AUTHENTICATION.FAILURE MODE]
          [if (NULL \LAFITE.ACTIVE.MODES)
            then ; Didn't get anywhere!
              [PRINTLAFITESTATUS (COND
                (\LAFITEMODE 'NO.MAILSERVER)
                (LAFITEMODELST 'MODE.NOT.SET)
                (T 'NO.MODE)
              )
              (RETURN NIL)
            ]
          else [for PAIR in FAILEDMODES do ; Show which modes failed
              (PRINTOUT PROMPTWINDOW T "Lafite " (fetch (LAFITEOPS
                                                         LAFITEMODE)
                                                         of (CDR PAIR))
                " mode suspended")
            ]
          ]

```







(LAFITE.STATUS.WITH.TIME

```
[LAMBDA (STR) ; Edited 9-May-88 15:41 by bvm
; Add current time to STR
(CONCAT STR " at " (DATE (DATEFORMAT NO.DATE NO.SECONDS CIVILIAN.TIME))
```

(LAFITE.REINITIALIZING

```
[LAMBDA (FIRSTTIME) ; Edited 9-May-88 15:51 by bvm
;; This guy alters status to show we're (re)initializing
(PRINTLAFITESTATUS (LAFITE.STATUS.WITH.TIME (if FIRSTTIME
; then "Initializing"
; else "Reinitializing"])
```

)
;; Parsing mail files

(DEFINEQ

(LAFITE.PARSE.FOLDER

```
[LAMBDA (FOLDER) ; Edited 1-May-89 14:35 by bvm
(LET* ((STREAM (\LAFITE.OPEN.FOLDER FOLDER 'INPUT :IGNORE))
(END (GETEOFPTR STREAM))
MESSAGES)
(COND
((OR (EQ END 0)
(SETQ MESSAGES (\LAFITE.PARSE.FOLDER1 FOLDER STREAM END 0 1)))
(replace (MAILFOLDER %#OFMESSAGES) of FOLDER with (COND
(MESSAGES (CAR MESSAGES))
(T 0)))
[replace (MAILFOLDER MESSAGEDESCRIPTORS) of FOLDER with (AND MESSAGES (\LAFITE.ADDMESSAGES.TO.ARRAY
NIL
(CDR MESSAGES)
1
(CAR MESSAGES)
(replace (MAILFOLDER TOCLASTMESSAGE#) of FOLDER with 0)
(replace (MAILFOLDER BROWSERREADY) of FOLDER with T)
FOLDER)
(T (\LAFITE.CLOSE.FOLDER FOLDER T)
NIL])
```

(LAFITE.PARSE.FOLDER1

```
[LAMBDA (FOLDER STREAM EOFPTR START FIRSTMSG# NOERROR) ; Edited 1-May-89 15:48 by bvm
(DECLARE (SPECVARS FOLDER STREAM EOFPTR START HERE LASTMSG)) ; Strictly for debugging
```

;;; Parse MAILFOLDER starting at byte START until end of file at EOFPTR. FIRSTMSG# is the ordinal to assign to the first message. Returns
;;; (lastmsg# . messagedescriptors), or NIL if there was any problem. If NOERROR is true, does not publicly complain about errors, but quietly returns
;;; NIL

```
(LAB.PROMPTPRINT FOLDER "Parsing " (COND
((EQ START 0)
"folder")
(T "additional msgs"))
' |...|)
(LET* ((HERE START)
(WINDOW (fetch (MAILFOLDER BROWSERPROMPTWINDOW) of FOLDER))
(XPOS (AND WINDOW (DSPXPOSITION NIL WINDOW)))
MSGLENGTH STAMPCOUNT MARK SEEN DELETED LASTMSG MINHEADER DUPLICATES)
(for MSG# from FIRSTMSG# while (< HERE EOFPTR)
collect (SETFILEPTR STREAM HERE)
;; the format of the stamp field of a laurel message is:
;; *start* <cr> <length of message in 5 ascii chars> <sp> <length of stamp in 5 ascii chars> <sp> <the char U or D> <the char
;; S or U> <any char> <cr>
;; U or D means Undeleted or Deleted; S or U means Seen or Unseen
[if (NOT (LA.READSTAMP STREAM))
then ; Length of previous message must be wrong
(RETURN (if (> HERE START)
then (BADMAILFILE FOLDER LASTMSG (SUB1 MSG#)
"Message length is incorrect" NOERROR $$VAL)
; First message is wrong
else (BADMAILFILE FOLDER LASTMSG MSG# "File does not appear to be in mail
format" NOERROR $$VAL)
[COND
([NOT (AND (SETQ MSGLENGTH (LA.READCOUNT STREAM))
(SETQ STAMPCOUNT (LA.READCOUNT STREAM))
(>= MSGLENGTH STAMPCOUNT))
; Read the 3 flag bytes
(PROGN (SETQ DELETED (SELECTC (BIN STREAM)
(UNDELETEDFLAG
NIL)
(DELETEDFLAG T)
(BADMAILFILE.FLAGBYTE FOLDER MSG#))
```

```

; read the U for Undeleted
(SETQ SEEN (SELECTC (BIN STREAM)
                   (UNSEENFLAG NIL)
                   (SEENFLAG T)
                   ((CHARCODE N)
                   ; For some reason, there are files with this for the Seen mark, so
                   ; allow it
                   T)
          (BADMAILFILE.FLAGBYTE FOLDER MSG#)))
; read the U for unseen
(SETQ MARK (BIN STREAM)) ; read the mark char
(EQ (BIN STREAM)
   (CHARCODE CR)))
(>= STAMPCOUNT (SETQ MINHEADER (- (GETFILEPTR STREAM)
                                  HERE])
; Expected to see msglength headerlength DSM<cr>.
(RETURN (BADMAILFILE FOLDER LASTMSG MSG# "Header is malformed" NOERROR $SVAL]
(PROG1 (SETQ LASTMSG (create LAFITEMSG
                            %# _ MSG#
                            BEGIN _ HERE
                            MESSAGELENGTH _ MSGLENGTH
                            MARKCHAR _ (OR (AND (NOT SEEN)
                                                UNSEENMARK)
                                           MARK)
                            SEEN? _ SEEN
                            DELETED? _ DELETED
                            STAMPLENGTH _ STAMPCOUNT))
[if (AND (> STAMPCOUNT MINHEADER)
        (EQ (BIN STREAM)
            (CHARCODE *)))
    then ; Duplicate junk?
        (LET ((INFO (CL:READ-LINE STREAM)))
            (if [AND (STRPOS "duplicate*" INFO 1 NIL T)
                    (FIXP (SETQ INFO (SUBATOM INFO 11))
                        then (push DUPLICATES (LIST INFO LASTMSG]
(LAFITE.PARSE.MSG.FOR.TOC LASTMSG FOLDER)
(add HERE MSGLENGTH)
(COND
  (XPOS (DSPXPOSITION XPOS WINDOW)
        (printout WINDOW .11 MSG#))))
finally (COND
  (XPOS ; Prepare to overwrite counter with 'done'
        (DSPXPOSITION XPOS WINDOW)))
(COND
  ((NOT (= HERE EOFPTR))
   [LAB.FORMAT FOLDER T "Warning: last message truncated from ~D to ~D bytes. "
    (fetch (LAFITEMSG MESSAGELENGTH) of LASTMSG)
    (replace (LAFITEMSG MESSAGELENGTH) of LASTMSG with (- (fetch (LAFITEMSG MESSAGELENGTH)
                                                                of LASTMSG)
                                                            (- HERE EOFPTR))
              (replace (LAFITEMSG MESSAGELENGTHCHANGED?) of LASTMSG with (replace (LAFITEMSG MARKSCHANGED?)
                                                                                   of LASTMSG with T))
              (replace (MAILFOLDER FOLDERNEEDSUPDATE) of FOLDER with T)))
  (if DUPLICATES
      then (\LAFITE.HANDLE.DUPLICATES FOLDER STREAM DUPLICATES $SVAL))
  (RETURN (CONS (fetch (LAFITEMSG %#) of LASTMSG)
                $SVAL]))

```

(\LAFITE.HANDLE.DUPLICATES

```

[LAMBDA (FOLDER FOLDERSTREAM DUPLICATES MESSAGES) ; Edited 2-May-89 12:02 by bvm
  (SORT DUPLICATES T) ; Sort by increasing file address
  (PROG* ((NEXT (CAR DUPLICATES))
          (NEXTMSG (CADR NEXT))
          FOUND NOTFOUND SCRATCH OLDSTART)
    [for OLDMSG in MESSAGES
      do (while (>= (SETQ OLDSTART (fetch (LAFITEMSG BEGIN) of OLDMSG))
                  (CAR NEXT))
          do (if (AND (= OLDSTART (CAR NEXT))
                    (\LAFITE.CHECK.DUPLICATE FOLDERSTREAM [OR SCRATCH (SETQ SCRATCH
                                                                    (OPENSTREAM "{nodircore}"
                                                                    'BOTH]
                                                                    OLDSTART
                                                                    (fetch (LAFITEMSG STAMPLENGTH) of OLDMSG)
                                                                    (fetch (LAFITEMSG MESSAGELENGTH) of OLDMSG)
                                                                    (fetch (LAFITEMSG BEGIN) of NEXTMSG)
                                                                    (fetch (LAFITEMSG STAMPLENGTH) of NEXTMSG)
                                                                    (fetch (LAFITEMSG MESSAGELENGTH) of NEXTMSG)))
                    then (push FOUND NEXTMSG)
                    else (push NOTFOUND NEXTMSG))
              (if (NULL (SETQ DUPLICATES (CDR DUPLICATES)))
                  then (GO DONE)
                  else (SETQ NEXTMSG (CADR (SETQ NEXT (CAR DUPLICATES)
                                                    DONE
                                                    (if FOUND
                                                        then (\LAFITE.REPORT.DUPLICATES FOLDER FOUND T))

```

```
(if NOTFOUND
  then (\LAFITE.REPORT.DUPLICATES FOLDER NOTFOUND)
    (for MSG in NOTFOUND when (fetch (LAFITEMSG DELETED?)) of MSG
      do
        ; Undelete these guys, since they may not really be duplicates
        ; any more
        (replace (LAFITEMSG DELETED?) of MSG with NIL)
        (replace (LAFITEMSG MARKSCHANGED?) of MSG with T)
        (replace (MAILFOLDER FOLDERNEEDSUPDATE) of FOLDER with T])
```

**(\LAFITE.CHECK.DUPLICATE**

```
[LAMBDA (FOLDERSTREAM SCRATCH START1 STAMP1 LENGTH1 START2 STAMP2 LENGTH2)
  ; Edited 2-May-89 11:58 by bvm
```

;; True if the messages in FOLDERSTREAM starting at START1 and START2 are identical, given their respective STAMP counts and LENGTHs.  
;; SCRATCH is a scratch stream to use for the comparison, since we can't have 2 file pointers in a stream at once.

```
(AND (= (- LENGTH1 STAMP1)
        (- LENGTH2 STAMP2))
  (PROGN (SETFILEPTR SCRATCH 0)
    (COPYBYTES FOLDERSTREAM SCRATCH (+ START1 STAMP1)
      (+ START1 LENGTH1))
    (SETFILEPTR SCRATCH 0)
    (SETFILEPTR FOLDERSTREAM (+ START2 STAMP2))
    (to (- LENGTH1 STAMP1) always (EQ (BIN SCRATCH)
      (BIN FOLDERSTREAM)]))
```

**(\LAFITE.REPORT.DUPLICATES**

```
[LAMBDA (FOLDER MESSAGES FOUNDP)
  ; Edited 1-May-89 15:52 by bvm
```

;; Return a string naming the messages in list NUMBERS, e.g., "Messages 23-39"

```
(LET ([DESCR (if (NULL (CDR MESSAGES))
  then (CL:FORMAT NIL "Message ~D" (fetch (LAFITEMSG %#) of (CAR MESSAGES)))
  else (LET [(NUMBERS (for MSG in MESSAGES collect (fetch (LAFITEMSG %#) of MSG)
    (if (for F in (CDR (SORT NUMBERS)) as I from (ADD1 (CAR NUMBERS))
      always (EQ F I))
      then
        ; Consecutive
        (CL:FORMAT NIL "Messages ~D--~D" (CAR NUMBERS)
          (CAR (LAST NUMBERS)))
      else
        ; Disconnected
        (CONCATLIST (CONS "Messages " (CDR (for F in NUMBERS
          join (LIST ", " F)]
    (ISARE (if (CDR MESSAGES)
      then "are"
      else "is"))))
  (LAB.FORMAT FOLDER (if FOUNDP
    then "~A are duplicates left over from an aborted Expunge."
    else "~A appear to be duplicates from an aborted Expunge, but the corresponding
      original messages were not found")
  DESCR])
```

**(BADMAILFILE**

```
[LAMBDA (FOLDER LASTMSG MSG# ERRSTR NOERROR MSGSOFAR)
  ; Edited 1-May-89 14:32 by bvm
```

```
(COND
  ((OR (NOT NOERROR)
    LAFITEDEBUGFLG)
  (LET* [(BROWSERWINDOW (fetch (MAILFOLDER BROWSERWINDOW) of FOLDER))
    (LASTLENGTHWRONG)
    (GOODPTR (AND LASTMSG (if (EQ MSG# (fetch (LAFITEMSG %#) of LASTMSG))
      then
        ; File only good up to previous message
        (SETQ LASTLENGTHWRONG T)
        (fetch (LAFITEMSG BEGIN) of LASTMSG)
      else
        ; Have a *start* here but header garbled
        (fetch (LAFITEMSG END) of LASTMSG]
    (CLEARW BROWSERWINDOW)
    (DSPRIGHTMARGIN (fetch (REGION RIGHT) of (DSPCLIPPINGREGION NIL BROWSERWINDOW))
      BROWSERWINDOW)
    ; In case it had been big from earlier
    (LINELENGTH T BROWSERWINDOW)
    (CL:FORMAT BROWSERWINDOW "Cannot parse file ~A ~%% at message ~D~@[, byte ~D~] because ~A~%%"
      (fetch (MAILFOLDER FULLFOLDERNAME) of FOLDER)
      MSG# GOODPTR ERRSTR)
  [COND
    (LASTMSG (printout BROWSERWINDOW T (if LASTLENGTHWRONG
      then "M"
      else "Last m")
      "message was:" T "Date: " (fetch (LAFITEMSG DATE) of LASTMSG)
      T "From: " (fetch (LAFITEMSG FROM) of LASTMSG)
      T "Subject: " (fetch (LAFITEMSG SUBJECT) of LASTMSG]
  (COND
    (NOERROR
      ; Only here for debugging intervention
      (HELP "Mail file parsing error" ERRSTR)
      NIL)
    (T
      ; Disable browser and prepare for scavenge
      (LAB.PROMPTPRINT FOLDER " Failed.")
      (LET* ((MENUW (fetch (MAILFOLDER BROWSERWINDOW) of FOLDER))
        [ITEMS ' ("Scavenge Folder" T "Attempt to repair the folder by scanning for message
```

```

boundaries")
("Just Close" NIL "Abort: Close this browser"]
(SCAVMENU (create MENU
ITEMS _ ITEMS
MENUFONT _ LAFITEMENUFONT
CENTERFLG _ T
MENURROWS _ 1
ITEMWIDTH _ (MAX (IQUOTIENT (WINDOWPROP BROWSERWINDOW 'WIDTH)
2)
(STRINGWIDTH (CAAR ITEMS)
LAFITEMENUFONT))
WHENSELECTEDFN _ [FUNCTION (LAMBDA (ITEM MENU BUTTON)
(LET
[(WINDOW (MAINWINDOW (WFROMMENU MENU)
(SHADEITEM ITEM MENU
LAFITEITEMBUSYSHADE)
(if (CADR ITEM)
then
; Do scavenge
(WINDOWPROP WINDOW
'DO-SCAVENGE T)
(WAKE.PROCESS
(WINDOWPROP WINDOW
'PROCESS))
else (CLOSEW WINDOW]
MENUBORDERSIZE _ 0
MENUOUTLINESIZE _ 0)))
(DELETEMENU (fetch (MAILFOLDER BROWSERMENU) of FOLDER)
NIL MENUW) ; Get rid of menu to avoid temptation
(replace (MAILFOLDER BROWSERSTATUS) of FOLDER with LAS.OUT.OF.DATE)
(WINDOWPROP BROWSERWINDOW 'PROCESS (THIS.PROCESS))
(WINDOWDELPROP BROWSERWINDOW 'CLOSEFN (FUNCTION LAB.CLOSEFN))
(WINDOWDELPROP BROWSERWINDOW 'SHRINKFN (FUNCTION LAB.SHRINKFN))
(WINDOWADDPROP BROWSERWINDOW 'CLOSEFN (FUNCTION BADMAILFILE.CLOSEFN))
(PRINTOUT BROWSERWINDOW T T "Select 'Scavenge' to attempt repair.")
(CL:UNWIND-PROTECT
(PROGN (ADDMENU SCAVMENU MENUW '(0 . 0))
(until (WINDOWPROP BROWSERWINDOW 'DO-SCAVENGE) do (BLOCK 60000)))
(WINDOWPROP BROWSERWINDOW 'PROCESS NIL)
(WINDOWDELPROP BROWSERWINDOW 'CLOSEFN (FUNCTION BADMAILFILE.CLOSEFN))
(if (NOT (WINDOWPROP BROWSERWINDOW 'DO-SCAVENGE))
then
; Getting rid of browser
(\LAFITE.FINISH.UPDATE BROWSERWINDOW FOLDER :EXIT)))
(if (AND (WINDOWPROP BROWSERWINDOW 'DO-SCAVENGE NIL)
(PROGN
; Run the scavenger
(DELETEMENU SCAVMENU NIL MENUW)
(replace (MAILFOLDER BROWSERSTATUS) of FOLDER with LAS.PARSING)
(WINDOWADDPROP BROWSERWINDOW 'CLOSEFN (FUNCTION LAB.CLOSEFN)
T)
(WINDOWADDPROP BROWSERWINDOW 'SHRINKFN (FUNCTION LAB.SHRINKFN)
T)
(\MAILSCAVENGE.INTERNAL FOLDER NIL GOODPTR MSG#)))
then
; Scavenge succeeded. Put browser back together
(ADDMENU (fetch (MAILFOLDER BROWSERMENU) of FOLDER)
MENUW
'(0 . 0))
[LET* ((NEWSTREAM (\LAFITE.OPEN.FOLDER FOLDER 'INPUT :IGNORE))
(NEWPARSE (\LAFITE.PARSE.FOLDER1 FOLDER NEWSTREAM (GETEOFPTR NEWSTREAM)
(OR GOODPTR 0)
MSG# T)))
(if NEWPARSE
then
; Return (last# . msgs)
(CONS (CAR NEWPARSE)
(NCONC (if LASTLENGTHWRONG
then
; Throw away the last message
(CL:NBUPTLAST MSGSOFAR)
else MSGSOFAR)
(CDR NEWPARSE]
else (LAB.PROMPTPRINT FOLDER " Aborted."))

```

(BADMAILFILE.CLOSEFN

```

[LAMBDA (WINDOW) ; Edited 3-May-89 19:16 by bvm
;; If user closes browser instead of selecting "scavenge" or "close", blow away the process waiting to scavenge
(PROCESS.EVAL (WINDOWPROP WINDOW 'PROCESS)
'(ERROR!))

```

(BADMAILFILE.FLAGBYTE

```

[LAMBDA (MAILFOLDER MSG#) (* bvm%: "24-Feb-86 12:08")
(LAB.PROMPTPRINT MAILFOLDER " [at msg " MSG# ": bad flag byte] ")
NIL])

```

(VERIFYMAILFOLDER

; Edited 23-Aug-88 15:47 by bvm

```
[LAMBDA (MAILFOLDER)
  (DECLARE (SPECVARS MSG# MSG HERE CHCUNT))
  [COND
    ((NOT (type? MAILFOLDER MAILFOLDER))
      (SETQ MAILFOLDER (\DTEST [COND
        ((WINDOWP MAILFOLDER)
          (WINDOWPROP MAILFOLDER 'MAILFOLDER))
        ((OR (LITATOM MAILFOLDER)
          (STRINGP MAILFOLDER))
          (LAFITE.OBTAIN.FOLDER MAILFOLDER 'INPUT]
          'MAILFOLDER]
      (WITH.MONITOR (fetch (MAILFOLDER FOLDERLOCK) of MAILFOLDER)
        (PROG (STREAM END)
          (SETQ STREAM (\LAFITE.OPEN.FOLDER MAILFOLDER 'INPUT :OK))
          [COND
            ((NOT (= (SETQ END (GETEOFPTR STREAM))
              (fetch (MAILFOLDER FOLDEREOFPTR) of MAILFOLDER)))
              (HELP "Length of file does not match Folder's idea of length" (LIST END]
            [bind CHCUNT STAMPCOUNT MARK MSG (HERE _ 0)
              (MESSAGES _ (fetch (MAILFOLDER MESSAGEDESCRIPTORS) of MAILFOLDER))
              (LASTMSG# _ (fetch (MAILFOLDER %#OFMESSAGES) of MAILFOLDER)) for MSG# from 1
              while (< HERE END) do (SETFILEPTR STREAM HERE)
                [COND
                  ((> MSG# LASTMSG#)
                    (RETURN (VERIFYFAILED MSG# "More messages in file than in core")
                      (SETQ MSG (NTHMESSAGE MESSAGES MSG#))
                    ;; the format of the stamp field of a laurel message is:
                    ;; *start* <cr> <length of message in 5 ascii chars> <sp> <length of stamp in 5 ascii chars> <sp> <the
                    ;; char U or D> <the char S or U> <any char> <cr>
                    ;; U or D means Undeleted or Deleted; S or U means Seen or Unseen
                    (COND
                      ((NOT (= (fetch (LAFITEMSG BEGIN) of MSG)
                        HERE))
                        (VERIFYFAILED MSG# "Message beginning pointer wrong"))
                      ((NOT (LA.READSTAMP STREAM))
                        (VERIFYFAILED MSG# "Bad Stamp"))
                      ([OR (NOT (SETQ CHCUNT (LA.READCOUNT STREAM)))
                        (NOT (= CHCUNT (fetch (LAFITEMSG MESSAGELENGTH) of MSG)
                          (VERIFYFAILED MSG# "Bad Message Length"))
                        ([OR (NOT (SETQ STAMPCOUNT (LA.READCOUNT STREAM)))
                          (NOT (= STAMPCOUNT (fetch (LAFITEMSG STAMPLength) of MSG)
                            (VERIFYFAILED MSG# "Bad Message Length"))
                          ((fetch (LAFITEMSG MARKSCHANGED?) of MSG))
                          ((NOT (EQ (SELECTC (BIN STREAM)
                            (UNDELETEDFLAG
                              NIL)
                              (DELETEDFLAG T)
                              '?))
                            (fetch (LAFITEMSG DELETED?) of MSG)))
                          (VERIFYFAILED MSG# "Disagreement in delete mark"))
                          ((NOT (EQ (SELECTC (BIN STREAM)
                            (UNSEENFLAG NIL)
                            (SEENFLAG T)
                            '?))
                            (fetch (LAFITEMSG SEEN?) of MSG)))
                          ; Figure out how to handle seen from me
                          (VERIFYFAILED MSG# "Disagreement in seen mark"))
                          ([NOT (OR (EQ (SETQ MARK (BIN STREAM)
                            (fetch (LAFITEMSG MARKCHAR) of MSG))
                            (NOT (fetch (LAFITEMSG SEEN?) of MSG)
                              (VERIFYFAILED MSG# "Disagreement in mark byte"))))
                          (add HERE CHCUNT)
                        finally (COND
                          ((NOT (= HERE END))
                            (VERIFYFAILED MSG# "Last message too short"]
                          (RETURN T))))])
  (RETURN T)))]
```

(VERIFYFAILED

; Edited 6-May-88 15:47 by bvm

```
[LAMBDA (MSG# ERRMSG)
  (HELP (CONCAT "Error in message " MSG# " : "
    ERRMSG])
```

(\LAFITE.READ.TOC.FILE

; Edited 2-Nov-89 15:58 by bvm  
; For \lafite.toceof

```
[LAMBDA (FOLDER TOCFILE)
  (DECLARE (SPECVARS FOLDER))
```

;;; Read table of contents file for FOLDER

;;; Header of toc file: <LafitePassword word> <LafiteVersion word> <EOF of mailfile integer> <last msg# in toc word>

;;; Body of toc file: one entry per message, of the form: <flags byte> <mark byte> <messagelength 3 bytes> <stamplength byte> <date> <subject  
ShortString> <From ShortString> <To ShortString>

;; Version history:

;; (8) Flags = 3 bits: parsed, deleted, seen. Date = 6 ascii bytes. Flag & Mark after stamp length.

;; (9) Flags = entire flag byte.

;; (10) Flag & Mark moved to front. Date conditional on flags: {if date fetched, then 4 bytes of idate} + {if ~date known, then 6 ascii bytes}. Also, msg length can be > 24 bits, in which case it is encoded as <255><4 bytes>. Not an incompatibility with older versions, since greatest msg length back then was 99999.

```
(LET [(TOCSTREAM (IGNORE-ERRORS (\LAFITE.OPENSTREAM (TOCFILENAME FOLDER)
'INPUT
'OLD
(FUNCTION \LAFITE.TOCEOF)])
(if TOCSTREAM
then (CL:UNWIND-PROTECT
(PROG (MSGCOUNTGUESS 0)
FOLDERSTREAM FOLDEREOPTR END MESSAGES EXTRAMESSES LASTMSG# READMORE TOCVERSION
PASSWORD)
(LAB.PROMPTPRINT FOLDER "Reading table of contents...")
(if (NEQ (SETQ PASSWORD (WORDIN TOCSTREAM))
LAFITETOCPASSWORD)
then (RETURN (BADTOCFILE FOLDER TOCSTREAM (if (EQ PASSWORD (LOGXOR 65535
LAFITETOCPASSWORD
))
; Caught during an Expunge
"inconsistent"
else "bad format"))))
elseif (< (SETQ TOCVERSION (WORDIN TOCSTREAM))
LAFITEVERSION#)
then (if (AND TOCVERSION (>= TOCVERSION 8))
then ; A slightly different format, still readable
(LAB.PROMPTPRINT FOLDER "(older format)")
else (RETURN (BADTOCFILE FOLDER TOCSTREAM "obsolete format")))
elseif (> TOCVERSION LAFITEVERSION#)
then ; New format, can't read
(RETURN (BADTOCFILE FOLDER TOCSTREAM "newer format"))
(SETQ FOLDERSTREAM (\LAFITE.OPEN.FOLDER FOLDER 'INPUT :IGNORE))
[if [NOT (= (SETQ END (FIXPIN TOCSTREAM))
(SETQ FOLDEREOPTR (GETEOPTR FOLDERSTREAM))
then ; Maybe new messages have been added to file
(SETFILEPTR FOLDERSTREAM END)
(if (NOT (LA.READSTAMP FOLDERSTREAM))
then (RETURN (BADTOCFILE FOLDER TOCSTREAM "inconsistent with mail
folder"))
else (SETQ READMORE T)
(SETQ MSGCOUNTGUESS (IQUOTIENT (- FOLDEREOPTR END)
500)
(add MSGCOUNTGUESS (SETQ LASTMSG# (WORDIN TOCSTREAM)))
(SETQ MESSAGES (\LAFITE.MAKE.MSGARRAY MSGCOUNTGUESS))
(for I from 1 to LASTMSG# bind MSG LENGTH (START _ (GETFILEPTR TOCSTREAM)
(MESSAGESTART _ 0)
do (SETQ MSG (create LAFITEMSG
%# _ I
BEGIN _ MESSAGESTART))
(if (>= TOCVERSION 10)
then ; These bytes are up front to make it easier to find them for
; update
(replace (LAFITEMSG MSGFLAGBITS) of MSG with (BIN TOCSTREAM))
(replace (LAFITEMSG MARKCHAR) of MSG with (BIN TOCSTREAM)))
;; Message length is 3 bytes long because it can be greater than MAX.SMALLP, though uncommon. In
;; fact, we even permit it to be greater than 24 bits, though that is exceedingly unlikely.
[replace (LAFITEMSG MESSAGELENGTH) of MSG
with (SETQ LENGTH (if (EQ (SETQ LENGTH (BIN TOCSTREAM))
0)
then ; Normal 16-bit length
(WORDIN TOCSTREAM)
else (\MAKENUMBER (if (EQ LENGTH 255)
then
; Very long length
(WORDIN TOCSTREAM)
else LENGTH)
(WORDIN TOCSTREAM)])
(add MESSAGESTART LENGTH)
(replace (LAFITEMSG STAMPLENGTH) of MSG with (BIN TOCSTREAM))
(if (< TOCVERSION 10)
then ; Flags and mark stuck in the middle here
(if (EQ TOCVERSION 8)
then (replace (LAFITEMSG PARSED&DELETED&SEENBITS) of MSG
with (BIN TOCSTREAM))
else (replace (LAFITEMSG MSGFLAGBITS) of MSG with (BIN TOCSTREAM)))
(replace (LAFITEMSG MARKCHAR) of MSG with (BIN TOCSTREAM))
; Clear datebits, since we once used a "formatted" bit in this area
(replace (LAFITEMSG DATEBITS) of MSG with 0)
elseif (fetch (LAFITEMSG DATEFETCHED?) of MSG)
then ; Read 4 bytes of idate
(\BINS TOCSTREAM MSG (UNFOLD (INDEXF (FETCH (LAFITEMSG IDATE)))
BYTESPERWORD))
```

```

4)
  (if (NOT (fetch (LAFITEMSG DATEKNOWN?) of MSG))
    then
      (replace (LAFITEMSG DATE) of MSG with (LA.READSTRING TOCSTREAM 6)))
  (replace (LAFITEMSG SUBJECT) of MSG with (LA.READSHORTSTRING TOCSTREAM))
  (replace (LAFITEMSG FROM) of MSG with (LA.READSHORTSTRING TOCSTREAM))
  (replace (LAFITEMSG TO) of MSG with (LA.READSHORTSTRING TOCSTREAM))
  (replace (LAFITEMSG TOCLENGTH) of MSG with (- (- START (SETQ START (GETFILEPTR
    TOCSTREAM)
    (SETA MESSAGES I MSG))
  (replace (MAILFOLDER TOCLASTMESSAGE#) of FOLDER with (if (EQ TOCVERSION LAFITEVERSION#)
    then LASTMSG#
    else
      ; Will have to rewrite toc next time
      0))
  [if READMORE
  then
    ; Parse new messages
    (if (SETQ EXTRAMESSAGES (\LAFITE.PARSE.FOLDER1 FOLDER FOLDERSTREAM
      FOLDEREOPTR END (ADD1 LASTMSG#)
      T))
      then [SETQ MESSAGES (\LAFITE.ADDMESSAGES.TO.ARRAY MESSAGES (CDR
        EXTRAMESSAGES
        )
        (ADD1 LASTMSG#)
        (SETQ LASTMSG# (CAR EXTRAMESSAGES)
      else (RETURN (BADTOCFILE FOLDER TOCSTREAM "Couldn't parse new messages,
        have to start from scratch..." T]
  (replace (MAILFOLDER MESSAGEDESCRIPTORS) of FOLDER with MESSAGES)
  (replace (MAILFOLDER %#OFMESSAGES) of FOLDER with LASTMSG#)
  (replace (MAILFOLDER FOLDEREOPTR) of FOLDER with FOLDEREOPTR)
  (replace (MAILFOLDER FOLDERNEEDSUPDATE) of FOLDER with NIL)
  (replace (MAILFOLDER BROWSERREADY) of FOLDER with T)
  (RETURN T))
  ;; Cleanups...
  (CLOSEF? TOCSTREAM)])

```

**(BADTOCFILE**

```

[LAMBDA (FOLDER TOCSTREAM ERRMSG CLEARFLG) ; Edited 1-May-89 14:26 by bvm
  (COND
    (CLEARFLG (LAB.PROMPTPRINT FOLDER T ERRMSG))
    (T (LAB.PROMPTPRINT FOLDER ERRMSG ", discarding...")))
  (COND
    (LAFITEDEBUGFLG (HELP "TOC file error" ERRMSG)))
    (DELFILE (CLOSEF TOCSTREAM))
    NIL]) ; Return NIL to tell loader to parse from scratch

```

**(\LAFITE.TOCEOF**

```

[LAMBDA (STREAM) ; Edited 2-Nov-89 15:58 by bvm

```

;;; Unexpected end of file on TOC, flush it

```

(RETFROM '\LAFITE.READ.TOC.FILE (BADTOCFILE FOLDER STREAM "Malformed table of contents, discarding...")

```

**(LA.READCOUNT**

```

[LAMBDA (STREAM NOVAL) ; Edited 21-Apr-89 14:49 by bvm
  ;; Read an integer terminated by a space. On success, return the integer, else NIL. If NOVAL is true, call is for test or effect, so we don't actually
  ;; compute the integer, but return NOVAL instead on success. File pointer is left following the terminating space.
  (bind VAL CH do (if (AND (<= (SETQ CH (BIN STREAM))
    (CHARCODE 9))
    (>= CH (CHARCODE 0)))
    then [SETQ VAL (OR NOVAL (+ (- CH (CHARCODE 0))
      (if VAL
        then (ITIMES 10 VAL)
        else 0]
    ; Non-digit, so done. Note that if we found no digits at all, VAL is
    ; NIL
    (RETURN (AND (EQ CH (CHARCODE SPACE))
      VAL])

```

**(LA.READSTAMP**

```

[LAMBDA (STREAM) (* bvm%: "22-Dec-83 18:23")
  (AND (EQ (BIN STREAM)
    (CHARCODE *))
    (EQ (BIN STREAM)
    (CHARCODE s))
    (EQ (BIN STREAM)
    (CHARCODE t))
    (EQ (BIN STREAM)
    (CHARCODE a))
    (EQ (BIN STREAM)
    (CHARCODE r))

```

```
(EQ (BIN STREAM)
    (CHARCODE t))
(EQ (BIN STREAM)
    (CHARCODE *))
(EQ (BIN STREAM)
    (CHARCODE CR))
```

**(LA.PRINTHEADER**

```
[LAMBDA (OUTPUTSTREAM BODYLENGTH MSG EXTRAHEADERLENGTH) ; Edited 25-Apr-89 12:39 by bvm
  ;; Print the header part of a Lafite msg up to but not including the flag/mark bytes, i.e., the part that reads ""start*<cr>nnnnn 000mm " for total
  ;; length nnnnn and header length mm. The message exclusive of the header is BODYLENGTH bytes long. If EXTRAHEADERLENGTH is given,
  ;; it tells how many bytes caller will write beyond the standard header. We will figure out the right header format to use. If MSG is supplied and its
  ;; STAMPLENGTH disagrees with the format we choose, we'll update it. Returns STAMPLENGTH.
  ;; Note that if the original message was from a Hardy file or other tool that sticks stuff in the header, we are now throwing that info away
  (POSITION OUTPUTSTREAM 0) ; So that LA.PRINTCOUNT doesn't screw up
  (LA.PRINTSTAMP OUTPUTSTREAM)
  (LET* ((TOTALLENGTH (+ BODYLENGTH LAFITESTAMPLENGTH))
        (FIELDWIDTH (if (< TOTALLENGTH 100000)
                        then ; We prefer to print the length as 5 digits, since that's Laurel/Hardy/MailTool format, but we'll use more if we
                        ; need to.
                        5
                        else (NCHARS TOTALLENGTH))))
    (STAMPLENGTH (+ FIELDWIDTH LAFITEBASICSTAMPLENGTH (OR EXTRAHEADERLENGTH 0)))
    FIELDDESC)
    [if (> STAMPLENGTH LAFITESTAMPLENGTH)
      then (add TOTALLENGTH (- STAMPLENGTH LAFITESTAMPLENGTH))
      ; Add the larger stamp's size to the total length. Problem: did
      ; this just make the total length overflow a power of 10, so we
      ; need another digit?
      (if (> (NCHARS TOTALLENGTH)
            FIELDWIDTH)
        then ; Yes, it's bigger than the space we allotted, so bump up by 1.
            ; This can't happen more than once (assuming
            ; EXTRAHEADERLENGTH is small), so no need to loop
            (add STAMPLENGTH 1)
            (add TOTALLENGTH 1)
            (add FIELDWIDTH 1))
        ; Now compute the format to print TOTALLENGTH
        (SETQ FIELDDESC `(FIX ,FIELDWIDTH 10 T]
        (LA.PRINTCOUNT TOTALLENGTH OUTPUTSTREAM FIELDDESC) ; total message length
        (LA.PRINTCOUNT STAMPLENGTH OUTPUTSTREAM) ; length of this header
        (if (AND MSG (NEQ (fetch (LAFITEMSG STAMPLENGTH) of MSG)
                            STAMPLENGTH))
            then ; Stamp size changed, so update MSG object
                (replace (LAFITEMSG STAMPLENGTH) of MSG with STAMPLENGTH)
                (replace (LAFITEMSG MESSAGELENGTH) of MSG with TOTALLENGTH))
            STAMPLENGTH])
```

**(LA.PRINTCOUNT**

```
[LAMBDA (COUNT STREAM FORMAT) ; Edited 17-Apr-89 16:48 by bvm
  [OR FORMAT (SETQ FORMAT '(FIX 5 10 T]
  (PRINTNUM FORMAT COUNT STREAM)
  (BOUT STREAM (CHARCODE SPACE]))
```

**(LA.PRINTSTAMP**

```
[LAMBDA (STREAM) (* bvm%: "27-Dec-83 12:54")
  (PROGN (BOUT STREAM (CHARCODE *))
        (BOUT STREAM (CHARCODE s))
        (BOUT STREAM (CHARCODE t))
        (BOUT STREAM (CHARCODE a))
        (BOUT STREAM (CHARCODE r))
        (BOUT STREAM (CHARCODE t))
        (BOUT STREAM (CHARCODE *))
        (BOUT STREAM (CHARCODE CR))
```

**(LA.READSHORTSTRING**

```
[LAMBDA (STREAM) ; Edited 10-Sep-87 14:21 by bvm:
```

;;; Read from STREAM a string written by LA.PRINTSHORTSTRING whose length is stored as the first byte.

```
(LET ((NBYTES (BIN STREAM)))
  (COND
    ((NEQ NBYTES 0)
     (LA.READSTRING STREAM NBYTES (if (EQ (\PEEKBIN STREAM)
                                           255)
                                       then ; a fat string. It is stored on the file in non-runcoded format.
                                       (BIN STREAM)
                                       T]))
```

**(LA.PRINTSHORTSTRING**

```
[LAMBDA (STREAM STRING) ; Edited 10-Sep-87 14:02 by bvm:
```



:: Store string on toc file. Format is: number of chars (as a byte), followed by chars. If string is fat, then chars are two bytes per char and are  
 :: prefixed by a 255 (impossible thin char).

```
(COND
  ((NULL STRING)
   (BOUT STREAM 0)
   1)
  (T (LET ((NBYTES (NCHARS STRING))
           (BASE (fetch (STRINGP BASE) of STRING))
           (OFF (fetch (STRINGP OFFST) of STRING)))
        (COND
          (> NBYTES 255) ; truncate string
          (SETQ NBYTES 255)))
      (BOUT STREAM NBYTES)
      (if (fetch (STRINGP FATSTRINGP) of STRING)
          then (BOUT STREAM 255)
              (\BOUITS STREAM BASE (UNFOLD OFF 2)
                (UNFOLD NBYTES 2))
              (+ 2 (UNFOLD NBYTES 2))
          else (\BOUITS STREAM BASE OFF NBYTES)
              (+ 1 NBYTES]))
```

**(LA.READSTRING**

[LAMBDA (STREAM NC FATP) ; Edited 10-Sep-87 14:22 by bvm:

:: Returns a string of length NC composed of the next NC (or 2\*NC if fatp) bytes of STREAM

```
(LET ((STR (ALLOCSTRING NC NIL NIL FATP)))
  (\BINS STREAM (fetch (STRINGP BASE) of STR)
   0
   (if FATP
        then (UNFOLD NC 2)
        else NC))
  STR))
```

**(LAFITE.VERIFYMSG**

[LAMBDA (MSG FOLDER) ; Edited 18-Jul-88 13:02 by bvm

:: Verify that this message starts points at an actual message start in the folder. If not, something is wrong with the toc--rebrowse, or take the  
 :: action given by \*LAFITE-VERIFY-ACTION\*.

```
(PROG ((STREAM (fetch (MAILFOLDER FOLDERSTREAM) of FOLDER)))
  (SETFILEPTR STREAM (fetch (LAFITEMSG BEGIN) of MSG))
  (COND
    ((NOT (LA.READSTAMP STREAM))
     (if *LAFITE-VERIFY-ACTION*
         then ; Caller anticipated this, and has provided an action
              (CL:FUNCALL *LAFITE-VERIFY-ACTION* MSG FOLDER STREAM)
         else (LET ((CHANGED (fetch (MAILFOLDER FOLDERNEEDSUPDATE) of FOLDER))
                  (ALLOW.BUTTON.EVENTS) ; Don't hoard mouse
                  (LAB.FORMAT FOLDER "Lafite's table of contents is inconsistent at message #~D; the
                    folder will have to be reparsed.~@[ However, you have unsaved changes.~]"
                  (fetch (LAFITEMSG %#) of MSG)
                  CHANGED)
                  (\LAFITE.REBROWSE.FOLDER FOLDER STREAM (OR CHANGED (AND LAFITEDEBUGFLG (HELP "TOC
                    inconsiste
                    nt"))))
                    NIL NIL :ABORT T]))
```

**(LA.MSGFROMMEP**

[LAMBDA (MSG) ; Edited 6-Jun-88 15:50 by bvm

:: True if the message is from the current user.

```
(AND \LAFITE.ACTIVE.MODES (PROG ((MODE (fetch (LAFITEMSG MODE) of MSG))
                                *LAFITE-MODE-DATA*)
  (if MODE
      then (if (NULL (SETQ *LAFITE-MODE-DATA* (\LAFITE.GET.USER.DATA MODE T)))
                then ; We know the mode, but haven't gotten authenticated yet, so
                     ; say NIL for now but be willing to change later
                (RETURN NIL))
            elseif (SETQ MODE (\LAFITE.GUESS.MODE MSG))
                  then (SETQ *LAFITE-MODE-DATA* (\LAFITE.GET.USER.DATA MODE T)))
      (RETURN (replace (LAFITEMSG MSGFROMMEP) of MSG
                      with (AND *LAFITE-MODE-DATA* (CL:FUNCALL (fetch (LAFITEMODEDATA
                                MESSAGE-FROM-SELF
                                )
                                of *LAFITE-MODE-DATA*)
                                MSG]))
```

)  
 (DEFINEQ

**(LAFITE.PARSE.MSG.FOR.TOC**

[LAMBDA (MSGDESCRIPTOR MAILFOLDER) ; Edited 25-Jun-91 14:25 by bvm

```
(COND
  ((NULL (fetch (LAFITEMSG PARSED?) of MSGDESCRIPTOR))
    (WITH.MONITOR (fetch (MAILFOLDER FOLDERLOCK) of MAILFOLDER)
      [PROG ((FOLDERSTREAM (\LAFITE.OPEN.FOLDER MAILFOLDER 'INPUT :ABORT))
        ORIGINAL-FROM)
        (for PAIR in (LAFITE.PARSE.HEADER FOLDERSTREAM \LAPARSE.TOCFIELDS (fetch (LAFITEMSG START)
          of MSGDESCRIPTOR)
            (fetch (LAFITEMSG END) of MSGDESCRIPTOR))
          do (SELECTQ (CAR PAIR)
            (From (replace (LAFITEMSG FROM) of MSGDESCRIPTOR with (CADR PAIR)))
            (Original-From
              (SETQ ORIGINAL-FROM (CADR PAIR)))
            (Subject (replace (LAFITEMSG SUBJECT) of MSGDESCRIPTOR with (CADR PAIR)))
            (Date (replace (LAFITEMSG DATE) of MSGDESCRIPTOR with (CADR PAIR)))
            (IDATE (replace (LAFITEMSG IDATE) of MSGDESCRIPTOR with (CADR PAIR)))
            (replace (LAFITEMSG DATEBITS) of MSGDESCRIPTOR with 3)
              ; Date fetched and known
            )
          NIL))
    (replace (LAFITEMSG PARSED?) of MSGDESCRIPTOR with T)
  [COND
    ((fetch (LAFITEMSG MSGFROMMEP) of MSGDESCRIPTOR)
      ; Get the TO field while we're at it, since TOC display will want it
      (LAFITE.FETCH.TO.FIELD MSGDESCRIPTOR MAILFOLDER)
    (COND
      ((AND LAFITEIFFFROMMETHENSEENFLG (NOT (fetch (LAFITEMSG SEEN?) of MSGDESCRIPTOR)))
        (replace (LAFITEMSG SEEN?) of MSGDESCRIPTOR with T)
        (replace (LAFITEMSG MARKCHAR) of MSGDESCRIPTOR with SEENMARK)
        (replace (LAFITEMSG MARKSCHANGED?) of MSGDESCRIPTOR with T]
      ;; Finally turn the From field into something more presentable. ORIGINAL-FROM is a perhaps more interesting address occurring on
      ;; messages via gateway. We didn't do this until now, because we wanted the mode to be guessed correctly.
      (replace (LAFITEMSG FROM) of MSGDESCRIPTOR with (LAFITE-EXTRACT-REAL-NAME
        (OR ORIGINAL-FROM (fetch (LAFITEMSG FROM)
          of MSGDESCRIPTOR)]))

```

**(LAFITE-EXTRACT-REAL-NAME**

```
[LAMBDA (FROMFIELD)
  ; Edited 6-May-92 09:48 by bvm
  ;; Called with (probably) an rfc822 From field. Return what we believe to be the user's "real name" for presentation purposes.
  (AND FROMFIELD (LET ((WHITESPACE ' (#\Space #\Tab))
    END OPEN CLOSE)
    [if (OR (CL:MEMBER (CL:CHAR FROMFIELD 0)
      WHITESPACE)
      (CL:MEMBER [CL:CHAR FROMFIELD (SETQ END (SUB1 (NCHARS FROMFIELD)
        WHITESPACE))
      then
        ; get rid of whitespace to avoid confusing us
        (SETQ FROMFIELD (CL:STRING-TRIM WHITESPACE FROMFIELD))
        (SETQ END (SUB1 (NCHARS FROMFIELD)
          WHITESPACE))
      (if (AND (SETQ OPEN (CL:POSITION #\< FROMFIELD))
        (> OPEN 0)
        (SETQ CLOSE (CL:POSITION #\> FROMFIELD :START OPEN))
        (EQ CLOSE END))
      then
        ; Real name <mail name>
        (CL:STRING-TRIM ' (#\Space #\Tab #\")
          (CL:SUBSEQ FROMFIELD 0 OPEN))
      elseif [AND (SETQ OPEN (CL:POSITION #\< FROMFIELD))
        (> OPEN 0)
        (SETQ CLOSE (CL:POSITION #\> FROMFIELD :START OPEN))
        (EQ CLOSE END)
        (NOT (CL:POSITION #\Space (CL:STRING-TRIM WHITESPACE (CL:SUBSEQ FROMFIELD 0
          OPEN))
      then
        ;; mail name (real name). Extra test is being careful about there maybe being more info in the rest of the
        ;; field
        (CL:STRING-TRIM WHITESPACE (CL:SUBSEQ FROMFIELD (ADD1 OPEN)
          CLOSE))
      elseif (AND (CL:POSITION #\: FROMFIELD)
        (NOT (CL:POSITION #\, FROMFIELD))
        (NOT (CL:POSITION #\; FROMFIELD))
        (SETQ OPEN (PARSE.NSNAME FROMFIELD)))
      then
        ;; A single ns name (the semi-colon test rules out rfc822 groups). Abbreviate it per local defaults. Note that
        ;; this requires reparsing the mail file if user changes domains, but that should be a rare event
        (NSNAME.TO.STRING OPEN)
      else FROMFIELD])

```

**(LAFITE.FETCH.TO.FIELD**

```
[LAMBDA (MSGDESCRIPTOR MAILFOLDER)
  ; Edited 23-Sep-87 18:35 by bvm
  ;; Fetch just the TO field of a message
  (OR (fetch (LAFITEMSG TO) of MSGDESCRIPTOR)
    (WITH.MONITOR (fetch (MAILFOLDER FOLDERLOCK) of MAILFOLDER)
      (replace (LAFITEMSG TO) of MSGDESCRIPTOR with (OR (LAFITE.PARSE.HEADER (\LAFITE.OPEN.FOLDER
```

```

MAILFOLDER
' INPUT :ABORT)
\ LAPARSE.TOFIELD
(fetch (LAFITEMSG START) of MSGDESCRIPTOR)
(fetch (LAFITEMSG END) of MSGDESCRIPTOR)
T)
UNSUPPLIEDFIELDSTR)))

```

(LAFITE.PARSE.HEADER

```

[LAMBDA (STREAM PARSETABLE START END ONCEONLY CHECKEOF) ; Edited 3-Feb-89 17:29 by bvm
(PROG (PARSERESULT PARSEBEGIN TABLE CH CHOICE)
(DECLARE (SPECVARS PARSERESULT PARSEBEGIN)) ; For Parse result functions to access
(COND
(START (SETFILEPTR STREAM START)))
TOP (SETQ TABLE PARSETABLE)
(SETQ PARSEBEGIN (GETFILEPTR STREAM))
LP [SELECTQ (CAR TABLE)
(CHOOSE (SETQ CH (UCASECODE (READCCODE STREAM)))
(COND
((find old CHOICE in (CDR TABLE) suchthat (EQ (CAR CHOICE)
CH))
(SETQ TABLE (CDR CHOICE))
(GO LP))))
(RESETQ TABLE (CDR TABLE))
(LAFITE.SKIP.WHITESPACE STREAM)
(COND
((OR (EQ (CL:FUNCALL (CAR TABLE)
STREAM
(CDR TABLE))
'STOP)
ONCEONLY)
(GO EXIT))
(T (GO NEXTLINE))))
(STOP [COND
((AND CHECKEOF (EQ CH (CHARCODE EOL)))
(push PARSERESULT (LIST 'EOF PARSEBEGIN)
(GO EXIT))
(COND
((EQ (SETQ CH (UCASECODE (READCCODE STREAM))
(CAR TABLE))
(SETQ TABLE (CDR TABLE))
(GO LP]

```

:: Get here if parse of current line failed

```

(COND
[CHECKEOF ; See if current line is end of header
(COND
([do (SELCHARQ CH ; Whitespace before a colon is illegal
((CR TAB SPACE)
(push PARSERESULT (LIST 'EOF PARSEBEGIN T))
(RETURN T))
(%: (LA.SKIP.TO.EOL STREAM CH)
(RETURN NIL))
(SETQ CH (READCCODE STREAM)
(GO EXIT)
(T (LA.SKIP.TO.EOL STREAM CH))
NEXTLINE
(COND
([COND
(END (< (GETFILEPTR STREAM)
END))
(T (NOT (\EOFP STREAM)
(GO TOP)))
EXIT
(replace CHARSET of STREAM with 0) ; Don't let any temporary change in charset affect future
; operations. This is not a call to CHARSET because of stupid
; bug that causes it to write a charset change!!!

(RETURN PARSERESULT])

```

(LAFITE.GRAB.DATE

```

[LAMBDA (STREAM) ; Edited 28-Apr-89 13:55 by bvm
(DECLARE (USEDFREE PARSERESULT))
(push PARSERESULT (LET ((DT (LAFITE.PARSE.DATE.FIELD STREAM))
(if (FIXP DT)
then (LIST 'IDATE DT)
else (LIST 'Date DT))

```

(LAFITE.READ.LINE.FOR.TOC

```

[LAMBDA (STREAM ARGS) (* bvm%: "19-Dec-83 14:08")
(DECLARE (USEDFREE PARSERESULT))
(PROG ((STR (LAFITE.READ.TO.EOL STREAM))
[COND
((IGREATERP (NCHARS STR)
255)

```

```
(SETQ STR (SUBSTRING STR 1 255 STR]
(push PARSERESULT (LIST (CAR ARGS)
                        STR]))
```

(LAFITE.READ.FORMAT

```
[LAMBDA (STREAM) ; (* bvm%: "12-Nov-84 17:21")
(DECLARE (USEDFREE PARSERESULT))
(PROG ((STR (LAFITE.READ.TO.EOL STREAM)))
(while (EQ (NTHCHARCODE STR -1)
           (CHARCODE SPACE))
  do (GLC STR)
(push PARSERESULT (LIST 'Format (MKATOM (U-CASE STR))
```

(LAFITE.READ.NAME.FIELD

```
[LAMBDA (STREAM ARGS) ; Edited 11-Jun-88 17:54 by bvm
(DECLARE (USEDFREE PARSERESULT))
;; For reading FROM, TO, etc. Just read the line, trim blanks, and return string. Can be more than one occurrence, so PARSERESULT value is a
;; list.
(PROG ((LINE (LAFITE.READ.TO.EOL STREAM)))
(do (SELCHARQ (NTHCHARCODE LINE -1)
            ((SPACE TAB %,) ; Strip off trailing spaces
            (GLC LINE))
    (RETURN NIL)))
(if (> (NCHARS LINE)
    0)
  then ; Ignore empty fields
    (for PAIR in PARSERESULT bind (FIELD _ (CAR ARGS)) when (EQ (CAR PAIR)
                                                                FIELD)
      do (RETURN (NCONC1 PAIR LINE)) finally (push PARSERESULT (LIST FIELD LINE))
```

(LAFITE.READ.ONE.LINE.FOR.TOC

```
[LAMBDA (STREAM) ; (* bvm%: "19-Dec-83 14:10")
(SETQ PARSERESULT (LAFITE.READ.TO.EOL STREAM))
```

(LAFITE.READ.TO.EOL

```
[LAMBDA (STREAM) ; Edited 22-Aug-88 16:24 by bvm
```

;;; Reads everything in STREAM up to next EOL and returns it as a string. If the next line starts with whitespace, it is assumed to be a continuation line, and it is returned as part of the result as well. See RFC 822

```
(PROG (RESULT LINE)
  LP (SETQ LINE (CL:READ-LINE STREAM))
  (SETQ RESULT (COND
                (RESULT ; EOL and leading whitespace are considered to be syntactically
                        ; a single space.
                (CONCAT RESULT " " LINE))
                (T LINE)))
  (SELCHARQ (PEEKCCODE STREAM T)
    ((SPACE TAB)
     (LAFITE.SKIP.WHITE.SPACE STREAM)
     (GO LP))
  NIL)
(RETURN RESULT])
```

(LA.SKIP.TO.EOL

```
[LAMBDA (STREAM LASTCH) ; Edited 22-Jun-2021 10:15 by rmk:
```

;;; Flush to end of this field. LASTCH is the last char read before this

```
(PROG* [(EOLC (fetch (STREAM EOLCONVENTION) of STREAM))
        (EOLCHAR (SELECTC EOLC
                        (LF.EOLC (CHARCODE LF))
                        (CHARCODE CR))
        (if (EQ LASTCH (CHARCODE EOL))
            then ; We're already there
              (GO PEEK))
  LP
  ;; Eat chars til eol.
  (repeatuntil (EQ EOLCHAR (\INCCODE STREAM)))
  (if (AND (EQ EOLC CRLF.EOLC)
           (EQ (\PEEKBIN STREAM T)
               (CHARCODE LF)))
      then ; Eat the lf after the cr
           ; \INCHAR would do that internally
            (\BIN STREAM))
  PEEK
  (SELCHARQ (\PEEKCCODE STREAM T)
    ((SPACE TAB) ; Continuation line, keep eating
     (GO LP))
  NIL])
```

(LAFITE.SKIP.WHITE.SPACE

; Edited 22-Jun-2021 10:18 by rmk:

```
[LAMBDA (STREAM)
  (do (SELCHARQ (\PEEKCCODE STREAM T)
      ((SPACE TAB)
       (\INCCODE STREAM))
      (RETURN]))
)
```

(DEFINEQ

(\LAFITE.PARSE.MESSAGE

; Edited 23-Sep-87 18:20 by bvm:

;; Return an alist of the header fields of MSGDESCRIPTOR specified by TABLE, which defaults to \LAPARSE.FULL. Aborts if folder has changed  
;; out from under.

```
(WITH.MONITOR (fetch (MAILFOLDER FOLDERLOCK) of MAILFOLDER)
  (LAFITE.PARSE.HEADER (\LAFITE.OPEN.FOLDER MAILFOLDER 'INPUT :ABORT)
   (OR TABLE \LAPARSE.FULL)
   (fetch (LAFITEMSG START) of MSGDESCRIPTOR)
   (fetch (LAFITEMSG END) of MSGDESCRIPTOR)))])
```

(RPAQQ LA.FULLPARSEFIELDS

```
(("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)))
```

(RPAQQ LA.TOCFIELDS

```
(("DATE:" LAFITE.GRAB.DATE)
 ("FROM:" LAFITE.READ.LINE.FOR.TOC From)
 ("SUBJECT:" LAFITE.READ.LINE.FOR.TOC Subject)
 ("ORIGINAL-FROM:" LAFITE.READ.LINE.FOR.TOC Original-From)))
```

(RPAQQ LA.TOFIELDONLY

```
(("TO:" LAFITE.READ.ONE.LINE.FOR.TOC)))
```

(RPAQQ LA.DATEFIELDONLY

```
(("DATE:" LAFITE.PARSE.DATE.FIELD.ONLY)))
```

(RPAQQ LA.SUBJECTFIELDONLY

```
(("SUBJECT:" LAFITE.READ.ONE.LINE.FOR.TOC)))
```

(DEFINEQ

(LAFITE.INIT.PARSE.TABLES

; Edited 26-Apr-89 14:09 by bvm

```
[LAMBDA NIL
  (SETQ \LAPARSE.FULL (LAFITE.MAKE.PARSE.TABLE LA.FULLPARSEFIELDS))
  (SETQ \LAPARSE.TOCFIELDS (LAFITE.MAKE.PARSE.TABLE LA.TOCFIELDS))
  (SETQ \LAPARSE.TOFIELD (LAFITE.MAKE.PARSE.TABLE LA.TOFIELDONLY))
  (SETQ \LAPARSE.SUBJECTFIELD (LAFITE.MAKE.PARSE.TABLE LA.SUBJECTFIELDONLY))
  (SETQ \LAPARSE.DATEFIELD (LAFITE.MAKE.PARSE.TABLE LA.DATEFIELDONLY))
```

(LAFITE.MAKE.PARSE.TABLE

; Edited 23-Sep-87 12:44 by bvm:

```
[LAMBDA (TABLE)
```

;;; Take a list of entries (string resultfn resultargs) and make a table usable by LAFITE.PARSE.HEADER

```
(LET ((PARSETABLE (LAFITE.MAKE.PARSE.TABLE1 (for ENTRY in TABLE collect (CONS (CL:STRING-UPCASE (CAR ENTRY))
                                                                              (CDR ENTRY))))
      (CONS 'CHOICE (NCONC PARSETABLE (CONSTANT `((, (CHARCODE CR)
                                                    STOP)
                                                    (, @ (CHARCODE
                                                         (* START *)))
                                                    STOP])))
```

(LAFITE.MAKE.PARSE.TABLE1

(\* bvm%: "30-Dec-83 11:12")

```
[LAMBDA (TABLE I)
```

;;; Subfunction of LAFITE.MAKE.PARSE.TABLE that builds a parsetable from the entries in TABLE splitting on character I

```
(PROG (ENTRY OTHERENTRIES DONE CHOICELIST CH)
  [for TAIL on TABLE unless (FMEMB (CAR TAIL)
                                     DONE)
  do (SETQ CH (NTHCHARCODE (CAR (SETQ ENTRY (CAR TAIL)))
                          I))
  (COND
   ((NULL CH)
    (ERROR (CAR ENTRY)
```

; Shouldn't happen: can't distinguish two them

```

" is an initial prefix of another entry"))
(push CHOICELIST (CONS CH (COND
  [[NOT (SETQ OTHERENTRIES (for X in (CDR TAIL) collect X
    when (EQ (NTHCHARCODE (CAR X)
      I)
      CH]
    ; This is the only choice
  (NCONC (for J from (ADD1 I) while (SETQ CH (NTHCHARCODE (CAR ENTRY)
    J))
    collect CH)
  (CONS 'RESULT (CDR ENTRY])
  (T (SETQ DONE (APPEND OTHERENTRIES DONE))
  (CONS 'CHOICE (LAFITE.MAKE.PARSE.TABLE1 (CONS ENTRY OTHERENTRIES
    )
    (ADD1 I]
  (RETURN CHOICELIST])
)

```

;; New header parser

(DEFINEQ

**(LAFITE.NEW.PARSE.HEADER**

```

[LAMBDA (STREAM PARSETABLE START END ONCEONLY CHECKEOF) ; Edited 6-Aug-93 18:18 by bvm
  (DECLARE (SPECVARS PARSETABLE PARSERESULT PARSEBEGIN))
  (PROG ([FIELD (OR *LAFITE-PARSE-HEADER-STRING-RESOURCE* (SETQ *LAFITE-PARSE-HEADER-STRING-RESOURCE*
    (ALLOCSTRING *LAFITE-MAX-FIELD-WIDTH*]
    PARSERESULT PARSEBEGIN CH I PATLEN) ; For Parse result functions to access
  (if START
    then (SETFILEPTR STREAM START))
  TOP (SETQ PARSEBEGIN (GETFILEPTR STREAM))
  (SETQ I 0)
  (do (SELCHARQ (SETQ CH (READCCODE STREAM))
    ((CR TAB SPACE NIL) ; Whitespace before a colon is illegal (or if it's a cr at start of line,
    ; it's the official end of header)
    (if CHECKEOF
      then (push PARSERESULT (LIST 'EOF PARSEBEGIN T)))
    [if (EQ CH (CHARCODE CR))
      then (for CHOICE in PARSETABLE when (EQ (CAR CHOICE)
        ',%)
        do ; Kludge for something to call at end of header
          (RETURN (CL:FUNCALL (CADR CHOICE)
            STREAM
            (CAR CHOICE)
            1
            (CDDR CHOICE]
          (GO EXIT))
    NIL)
  (if (< I *LAFITE-MAX-FIELD-WIDTH*)
    then (CL:SETF (CL:CHAR FIELD I)
      (CL:CODE-CHAR CH))
    (add I 1))
  (if (EQ CH (CHARCODE ":"))
    then [for CHOICE in PARSETABLE when (AND (<= (SETQ PATLEN (NCHARS (CAR CHOICE)))
      I)
      (STRING-EQUAL FIELD (CAR CHOICE)
        :END1 PATLEN))
    do (LAFITE.SKIP.WHITE.SPACE STREAM)
      (COND
        ((OR (EQ (CL:FUNCALL (CADR CHOICE)
          STREAM FIELD I (CDDR CHOICE))
          'STOP)
          ONCEONLY)
          (GO EXIT))
        (T (GO NEXTLINE]
    ;; Get here if parse of current line failed
    (LA.SKIP.TO.EOL STREAM CH)
    (GO NEXTLINE)))
  NEXTLINE
  (COND
    ([COND
      (END (< (GETFILEPTR STREAM)
        END))
      (T (NOT (\EOFP STREAM)
        (GO TOP)))
    EXIT
  (replace CHARSET of STREAM with 0) ; Don't let any temporary change in charset affect future
  ; operations. This is not a call to CHARSET because of stupid
  ; bug that causes it to write a charset change!!!
  (RETURN PARSERESULT])
)

```

**(LAFITE.HANDLE.ORIGINAL.FIELD**

```

[LAMBDA (STREAM FIELD FIELDLEN IGNORE)

```

```

(DECLARE (USEDFREE PARSERESULT PARSEBEGIN PARSETABLE)) ; Edited 3-Jun-92 17:51 by bvm
;; Called when we parsed a header starting "Original-xxx:..." We want to hide the "Original-" part, and also hide the matching "xxx:" field that (we
;; assume) occurs later
(LA.SKIP.TO.EOL STREAM)
[push PARSERESULT (LIST PARSEBEGIN (+ PARSEBEGIN (CONSTANT (NCHARS "Original-"))
(push PARSETABLE (LIST (CL:SUBSEQ FIELD (CONSTANT (NCHARS "Original-"))
FIELDLEN)
(FUNCTION LAFITE.EAT.UNDESIRABLE.FIELD)))
; Note that we have to COPY the characters of field, since that
; string is volatile
NIL])
)
(RPAQ? *LAFITE-MAX-FIELD-WIDTH* 100)
(RPAQ? *LAFITE-PARSE-HEADER-STRING-RESOURCE* )
(RPAQ? *LAFITE-VERIFY-ACTION* NIL)
(RPAQ? MAILWATCHWAITTIME 5)
(RPAQ? LAFITEFLUSHMAILFLG T)
(RPAQ? LAFITETOC.EXT "-Lafite-toc")
(RPAQ? LAFITENEWMAILFN NIL)
(RPAQ? LAFITENEWMILTUNE NIL)
(RPAQ? LAFITEGETMAILTUNE NIL)
(RPAQ? LAFITE.AFTER.GETMAIL.FN NIL)
(RPAQ? LAFITE.SORT.NEW.MAIL NIL)
(RPAQ? \LAFITE.LAST.STATUS )
(ADDTOVAR \SYSTEMCACHEVARS \LAFITE.LAST.STATUS)
(DECLARE%: DOEVAL@COMPILE
(CL:PROCLAIM '(CL:SPECIAL *LAFITE-VERIFY-ACTION* DEFAULTREGISTRY LAFITEDEBUGFLG))
(CL:PROCLAIM '(GLOBAL LAFITEFLUSHMAILFLG LAFITEGETMAILTUNE LAFITENEWMAILFN LAFITENEWMILTUNE
LAFITEIFFFROMMETHENSEENFLG MAILWATCHWAITTIME LAFITETOC.EXT LAFITE.SORT.NEW.MAIL))
)
(DECLARE%: EVAL@COMPILE DONTCOPY
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS *LAFITE-MAX-FIELD-WIDTH* *LAFITE-PARSE-HEADER-STRING-RESOURCE* LA.FULLPARSEFIELDS LA.TOCFIELDS
LA.TOFIELDONLY LA.DATEFIELDONLY LA.SUBJECTFIELDONLY \LAFITE.AUTHENTICATION.FAILURE \LAPARSE.FULL
\LAPARSE.TOCFIELDS \LAPARSE.TOFIELD \LAPARSE.SUBJECTFIELD \LAPARSE.DATEFIELD LAFITE.AFTER.GETMAIL.FN)
)
(FILESLOAD (SOURCE)
LAFITE-DECLS)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
)

```

---

FUNCTION INDEX

BADMAILFILE .....	11	LAFITE.MAKE.PARSE.TABLE1 .....	21	\LAFITE.GETMAIL.FROM.ICON .....	1
BADMAILFILE.CLOSEFN .....	12	LAFITE.NEW.PARSE.HEADER .....	22	\LAFITE.GETMAIL.PROC .....	1
BADMAILFILE.FLAGBYTE .....	12	LAFITE.PARSE.HEADER .....	19	\LAFITE.GETNEWMAIL .....	2
BADTOCFILE .....	15	LAFITE.PARSE.MSG.FOR.TOC .....	17	\LAFITE.GETNEWMAIL# .....	3
FULLUSERNAME .....	5	LAFITE.READ.FORMAT .....	20	\LAFITE.GETNEWMAIL1 .....	2
LA.MSGFROMMEP .....	17	LAFITE.READ.LINE.FOR.TOC .....	19	\LAFITE.GUESS.MODE .....	5
LA.PRINTCOUNT .....	16	LAFITE.READ.NAME.FIELD .....	20	\LAFITE.HANDLE.DUPLICATES .....	10
LA.PRINTHEADER .....	16	LAFITE.READ.ONE.LINE.FOR.TOC .....	20	\LAFITE.NEW.MAIL.EXISTS .....	8
LA.PRINTSHORTSTRING .....	16	LAFITE.READ.TO.EOL .....	20	\LAFITE.PARSE.FOLDER .....	9
LA.PRINTSTAMP .....	16	LAFITE.SKIP.WHITE.SPACE .....	21	\LAFITE.PARSE.FOLDER1 .....	9
LA.READCOUNT .....	15	LAFITE.STATUS.WITH.TIME .....	9	\LAFITE.PARSE.MESSAGE .....	21
LA.READSHORTSTRING .....	16	LAFITE.USER.NAME.FROM.LOGIN .....	5	\LAFITE.READ.TOC.FILE .....	13
LA.READSTAMP .....	15	LAFITECLEARCACHE .....	5	\LAFITE.REGISTER.MODE .....	5
LA.READSTRING .....	17	LAFITEMAILWATCH .....	6	\LAFITE.REINITIALIZING .....	9
LA.SKIP.TO.EOL .....	20	POLLNEWMAIL .....	6	\LAFITE.REPORT.DUPLICATES .....	11
LAFITE-EXTRACT-REAL-NAME .....	18	PRINTLAFITESTATUS .....	8	\LAFITE.RETRIEVEMESSAGES .....	3
LAFITE.FETCH.TO.FIELD .....	18	VERIFYFAILED .....	13	\LAFITE.TOCEOF .....	15
LAFITE.GRAB.DATE .....	19	VERIFYMAILFOLDER .....	12	\LAFITE.VERIFYMSG .....	17
LAFITE.HANDLE.ORIGINAL.FIELD .....	22	\LAFITE.CHECK.DUPLICATE .....	11	\LAFITE.WAKE.WATCHER .....	6
LAFITE.INIT.PARSETABLES .....	21	\LAFITE.GET.USER.DATA .....	4		
LAFITE.MAKE.PARSE.TABLE .....	21	\LAFITE.GETMAIL .....	1		

---

VARIABLE INDEX

*LAFITE-MAX-FIELD-WIDTH* .....	23	LAFITE.SORT.NEW.MAIL .....	23
*LAFITE-PARSE-HEADER-STRING-RESOURCE* .....	23	LAFITEFLUSHMAILFLG .....	23
*LAFITE-VERIFY-ACTION* .....	23	LAFITEGETMAILTUNE .....	23
LA.DATEFIELDONLY .....	21	LAFITENEWMMAILFN .....	23
LA.FULLPARSEFIELDS .....	21	LAFITENEWMMAILTUNE .....	23
LA.SUBJECTFIELDONLY .....	21	LAFITETOC.EXT .....	23
LA.TOCFIELDS .....	21	MAILWATCHWAITTIME .....	23
LA.TOFIELDONLY .....	21	\LAFITE.LAST.STATUS .....	23
LAFITE.AFTER.GETMAIL.FN .....	23	\SYSTEMCACHEVARS .....	23

---