

File created: 1-Feb-2022 17:06:07 {DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>MTP.;2

changes to: (VARS MTPCOMS)
(FNS MTP.MAKEANSWERFORM)

previous date: 19-May-86 16:54:58 {DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>MTP.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1983-1984, 1986 by Xerox Corporation.

(RPAQQ MTPCOMS

```
((COMS (* Lafite mode MTP)
  (FNS MTP.GET.USERDATA MTP.DELIVERMESSAGE MTP.PREPARE.SEND MTP.MAKEANSWERFORM)
  (ADDVARS (LAFITEMODELST (MTP MTP.PREPARE.SEND MTP.DELIVERMESSAGE MTP.MAKEANSWERFORM
                        MTP.GET.USERDATA)))
  (FNS \MTP.AUTHENTICATE \MTP.COERCE.MSG \MTP.FILL \MTP.INDENT \MTP.CLRBUF \MTP.PRINTADDRESSES)
  (INITVARS (MTP.SERVER)
    (MTP.LINELENGTH 70)
    (MTP.RIGHTMARGINWIDTH 10)
    (MTP.FILLMSGFLG %'ASK)
    (MTP.INSERTANSWERFLG T)
    (MTP.INSERTANSWERNSPACES 3)))
[COMS (* MTP mail server)
  (FNS MTP.OPENMAILBOX MTP.POLLNEWMAIL MTP.NEXTMESSAGE MTP.RETRIEVEMESSAGE MTP.CLOSEMAILBOX)
  (FNS \MTP.ENDOFMESSAGESTATE \MTP.POLLNEWMAIL)
  (ADDVARS (MAILSERVERTYPES (MTP MTP.POLLNEWMAIL MTP.OPENMAILBOX MTP.NEXTMESSAGE MTP.RETRIEVEMESSAGE
                            MTP.CLOSEMAILBOX ETHERPORT))
  (FILES LAFITE)
  (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS MTPMAILBOX MTPPARSE)
    (CONSTANTS \PUPSOCKET.MTP \PUPSOCKET.MISCSERVICES)
    (CONSTANTS * PUPTYPES)
    (GLOBALVARS MTP.SERVER MTP.LINELENGTH MTP.RIGHTMARGINWIDTH MTP.FILLMSGFLG MTP.INSERTANSWERFLG
      MTP.INSERTANSWERNSPACES \LAPARSE.FULL LAFITEEDITORFONT UNSUPPLIEDFIELDSTR MESSAGESTR
      \LAFITEUSERDATA MAILSERVERTYPES \LAFITE.AUTHENTICATION.FAILURE)
    (FILES (LOADCOMP)
      LAFITE DPUPFTP))))
```

(* * Lafite mode MTP)

(DEFINEQ

(MTP.GET.USERDATA

```
[LAMBDA NIL (* drc%: "29-Apr-86 23:31")
  (LET ((PORT (ETHERPORT MTP.SERVER))
        USER/PWD)
    (SETQ \LAFITEUSERDATA
      (if (NULL PORT)
        then (PRINTOUT PROMPTWINDOW T "MTP.SERVER not found -- " MTP.SERVER T)
              (SETQ \LAFITE.AUTHENTICATION.FAILURE "No Server")
              NIL
        else (SETQ USER/PWD (\INTERNAL/GETPASSWORD MTP.SERVER))
              (AND (MTP.AUTHENTICATE MTP.SERVER USER/PWD)
                (create LAFITEUSERDATA
                  FULLUSERNAME _ (CAR USER/PWD)
                  ENCRYPTEDPASSWORD _ (CDR USER/PWD)
                  SHORTUSERNAME _ (CAR USER/PWD)
                  MAILSERVERS _ (LIST (create MAILSERVER
                    MAILPORT _ PORT
                    MAILSERVERNAME _ MTP.SERVER
                    MAILSERVEROPS _ (CDR (ASSOC %'MTP MAILSERVERTYPES]))
```

(MTP.DELIVERMESSAGE

```
[LAMBDA (MSG PARSE W ABORTW) (* drc%: "29-Apr-86 23:38")
  (DECLARE (GLOBALVARS FTPDEBUGFLG FTPDEBUGLOG))
  (RESETLST
    (LET* ((USERDATA (\LAFITE.GET.USER.DATA))
           (USER (fetch (LAFITEUSERDATA FULLUSERNAME) of USERDATA))
           (MAILSERVER (CAR (fetch (LAFITEUSERDATA MAILSERVERS) of USERDATA)))
           [PLIST (LIST (LIST %'MAILBOX (fetch (MTPPARSE MAILBOX) of PARSE))
                       (LIST %'SENDER (CONCAT USER "@" (fetch MAILSERVERNAME of MAILSERVER))
                       (PW (GETPROMPTWINDOW W))
                       (TEXT (\MTP.COERCE.MSG MSG (fetch (MTPPARSE EOH) of PARSE)
                         PW))
                       INS OUTS)
           (AND (WINDOWPROP ABORTW %'ABORT)
                (ERROR!))
           (PRINTOUT PW "delivering...")
           (SETQ INS (OPENBSPSTREAM (CONS (CAR (fetch (MAILSERVER MAILPORT) of MAILSERVER))
```

```

                                \PUPSOCKET.MTP)
                                NIL %'\FTP.ERRORHANDLER))
(if INS
  then (RESETSAVE NIL (LIST %'CLOSEBSPSTREAM INS 5000))
  else (PRINTOUT PW (fetch (MAILSERVER MAILSERVERNAME) of MAILSERVER)
        " not responding. ")
      (ERROR!))
(SETQ OUTS (BSPOUTPUTSTREAM INS))
(FTPPUTMARK OUTS (MARK# STORE-MAIL))
(\FTP.PRINTPLIST OUTS PLIST)
(FTPPUTMARK OUTS (MARK# EOC))
(SELECTC (FTPGETMARK INS)
  ((MARK# YES)
   (FTPGETCODE INS)
   (\FTP.FLUSH.TO.EOC INS (AND FTPDEBUGFLG FTPDEBUGLOG)))
  ((MARK# NO)
   (FTPGETCODE INS)
   (\FTP.FLUSH.TO.EOC INS PW)
   (ERROR!))
  (\FTPERROR INS))
(FTPPUTMARK OUTS (MARK# HERE-IS-FILE))
(PRINTOUT OUTS (fetch (MTPPARSE FROMLINE) of PARSE)
  T)
(PRINTOUT OUTS (fetch (MTPPARSE DATELINE) of PARSE)
  T)
(COPYBYTES TEXT OUTS)
(if (WINDOWPROP ABORTW %'ABORT)
  then (FTPPUTMARK OUTS (MARK# NO))
  else (FTPPUTMARK OUTS (MARK# YES)))
(FTPPUTMARK OUTS (MARK# EOC))
(SELECTC (FTPGETMARK INS)
  ((MARK# YES)
   (FTPGETCODE INS)
   (\FTP.FLUSH.TO.EOC INS (AND FTPDEBUGFLG FTPDEBUGLOG)))
  (PROGN (FTPGETCODE INS)
   (\FTP.FLUSH.TO.EOC INS PROMPTWINDOW)
   (ERROR!)))
T)))]

```

(MTP.PREPARE.SEND

(* drc%: "17-May-86 17:34")

```

[LAMBDA (MSG W)
  (LET* [(PARSE (\LAFITE.PREPARE.SEND MSG W))
        (RECIPIENTS (APPEND (CDR (FASSOC %'To PARSE))
                             (CDR (FASSOC %'cc PARSE))
                             (OR PARSE (\SENDMESSAGEFAIL W "Bad message format.")
                                     (AND (FASSOC %'Sender PARSE)
                                         (\SENDMESSAGEFAIL W "Can't specify Sender!"))
                                     (AND (FASSOC %'Date PARSE)
                                         (\SENDMESSAGEFAIL W "Can't specify Date!"))
                                     (OR RECIPIENTS (\SENDMESSAGEFAIL W "No recipients?"))
                                     (create MTPPARSE
                                       FROMLINE _ (CONCAT (if (ASSOC %'From PARSE)
                                                             then "Sender: "
                                                             else "From: ")
                                                           (FULLUSERNAME))
                                       MAILBOX _ [CONCATLIST (for TAIL on RECIPIENTS collect (if (CDR TAIL)
                                                                                             then (CONCAT (CAR TAIL)
                                                                                             ", ")
                                                                                             else (CAR TAIL))]
                                       EOH _ (CADR (FASSOC %'EOF PARSE))
                                       DATELINE _ (CONCAT "Date: " (DATE (DATEFORMAT DAY.OF.WEEK SPACES TIME.ZONE NO.SECONDS)))]

```

(MTP.MAKEANSWERFORM

; Edited 1-Feb-2022 17:05 by rmk
(* drc%: "19-May-86 15:39")

```

[LAMBDA (MSGS FOLDER)
  (PROG ((OLD.MSG (OR (CAR (LISTP MSGS))
                     MSGS))
        [INSERT? (AND MTP.INSERTANSWERFLG (MENU (\LAFITE.CREATE.MENU %' ("Yes" T "Insert the text of the
                                                                    message being answered")
                                                                    ("No" NIL "Normal answer form")
                                                                    ("Abort" %'ABORT "Abort Answer
                                                                    command"))
                                                                    "Insert Message?")
        (OLD.TEXT (\LAFITE.OPEN.FOLDER FOLDER %'INPUT))
        START END OLD.FIELDS SUBJECT FROM TO CC DATE REPLY-TO SENDER NEW.MSG NEW.TO NEW.CC)
        (if (EQ INSERT? %'ABORT)
          then (RETURN))
        (SETQ START (fetch (LAFITEMSG START) of OLD.MSG))
        (SETQ END (fetch (LAFITEMSG END) of OLD.MSG))
        (SETQ OLD.FIELDS (LAFITE.PARSE.HEADER OLD.TEXT \LAPARSE.FULL START END))
        (for PAIR in OLD.FIELDS do (SELECTQ (CAR PAIR)
          (Subject (SETQ SUBJECT (CADR PAIR)))
          (From (SETQ FROM (CDR PAIR)))
          (To (SETQ TO (CDR PAIR)))

```

```

(cc (SETQ CC (CDR PAIR)))
(Date (SETQ DATE (CADR PAIR)))
(Reply-to (SETQ REPLY-TO (CDR PAIR)))
(Sender (SETQ SENDER (CDR PAIR)))
(NIL))
(SETQ NEW.TO (OR REPLY-TO FROM SENDER))
(OR NEW.TO (RETURN (LAB.PROMPTPRINT FOLDER "Can't reply -- no From or Sender")))
(SETQ NEW.MSG (OPENTEXTSTREAM NIL NIL NIL NIL (LIST %'FONT LAFITEEDITORFONT)))
(LINELENGTH MAX.SMALLP NEW.MSG)
(PRINTOUT NEW.MSG "Subject: ")
(if (NOT (STRING-EQUAL (SUBSTRING SUBJECT 1 3)
"Re:"))
then (printout NEW.MSG "Re: "))
(PRINTOUT NEW.MSG (OR SUBJECT UNSUPPLIEDFIELDSTR
T))
(AND FROM (PRINTOUT NEW.MSG "In-reply-to: " (CAR FROM)
's message of " DATE T))
(PRINTOUT NEW.MSG "To: ")
(MTP.PRINTADDRESSES NEW.TO NEW.MSG)
(SETQ NEW.CC (LA.SETDIFFERENCE (if REPLY-TO
then (LIST (FULLUSERNAME))
else (LA.REMOVEDUPLICATES (APPEND TO CC)))
NEW.TO))
(if NEW.CC
then (PRINTOUT NEW.MSG "cc: ")
(MTP.PRINTADDRESSES NEW.CC NEW.MSG))
(TERPRI NEW.MSG)
(if INSERT?
then (MTP.FILL OLD.TEXT NEW.MSG MTP.INSERTANSWERNSPACES MTP.LINELENGTH START END)
(PRINTOUT NEW.MSG MESSAGESTR T)
else (LET [(SELECTPOSITION (ADD1 (GETFILEPTR NEW.MSG)
(PRINTOUT NEW.MSG MESSAGESTR T)
(TEDIT.SETSEL NEW.MSG SELECTPOSITION (NCHARS MESSAGESTR)
%'RIGHT T)))
(RETURN NEW.MSG)])

```

```

)
(ADDTOVAR LAFITEMODELST (MTP MTP.PREPARE.SEND MTP.DELIVERMESSAGE MTP.MAKEANSWERFORM MTP.GET.USERDATA))
(DEFINEQ

```

(MTP.AUTHENTICATE

```

[LAMBDA (HOST USER/PWD) (* drc%: "25-Apr-86 13:06")
```

(* I couldn't get PUP authentication to work w/ our Misc server, so we just check for mailbox existence. Password checking is done when retrieving mail.)

```

(LET* ((RESPONSE (MTP.POLLNEWMAIL HOST (CAR USER/PWD)))
(TYPE (CAR RESPONSE))
(MESSAGE (CDR RESPONSE)))
(SELECTC TYPE
((LIST \PT.NEWMAIL \PT.NONEWMAIL)
T)
((LIST \PT.NOMAILBOX \PT.ERROR)
(SETQ \LAFITE.AUTHENTICATION.FAILURE MESSAGE)
NIL)
(NIL (PRINTOUT PROMPTWINDOW T HOST " not responding to authentication request." T)
(SETQ \LAFITE.AUTHENTICATION.FAILURE "No Server")
NIL)
NIL))

```

(MTP.COERCE.MSG

```

[LAMBDA (MSG EOH ECHOSTREAM) (* drc%: "19-May-86 16:08")
```

```

(DECLARE (GLOBALVARS MTP.LINELENGTH))
(LET [(STREAM (COERCETEXTOBJ MSG %'STREAM))
(FILL? (SELECTQ MTP.FILLMSGFLG
(ALWAYS T)
(ASK (MENU (\LAFITE.CREATE.MENU %' ("Yes" T "Break long lines in message to
MTP.LINELENGTH")
("No" NIL "Deliver message as is")
("Abort" %'ABORT "Abort deliver command"))
"Fill Text?"))))
(NEVER NIL)
(SHOULDNT)]
(if (EQ FILL? %'ABORT)
then (ERROR!))
(if FILL?
then (PRINTOUT ECHOSTREAM "filling...")
(LET ((OUTS (OPENSTREAM %' {NODIRCORE} %'BOTH))
(COPYBYTES STREAM OUTS 0 EOH)
(MTP.FILL STREAM OUTS 0 MTP.LINELENGTH)
(SETFILEPTR OUTS 0)
OUTS)
else STREAM])

```

(\MTP.FILL

[LAMBDA (INS OUTS LMARGIN RMARGIN START END) (* drc%: "19-May-86 16:46")

(* * Copy bytes from INS to OUTS, indenting to LMARGIN. New lines started at last space before RMARGIN -- unless the line ends before RMARGIN + MTP.RIGHTMARGINWIDTH anyway. Copy from START (default is current pos) to END (default is EOF)%.)

(until (GEQ (GETFILEPTR INS) END)

as COLUMN from (ADD1 LMARGIN) bind (LINEBUF _ (OPENSTREAM %' {NODIRCORE} %' BOTH)) (CARRY _ LMARGIN) (END _ (OR END (GETEOFPTR INS))) (LIMIT _ (IPLUS RMARGIN MTP.RIGHTMARGINWIDTH)) (EDGE _ (ADD1 RMARGIN)) BYTE SPACE SPACES

first (AND START (SETFILEPTR INS START)) (\MTP.INDENT INS OUTS END LMARGIN)

eachtime (SETQ BYTE (BIN INS)) (SELCHARQ BYTE) ((SPACE TAB) (BOUT LINEBUF BYTE) (push SPACES COLUMN)) (EOL (SETFILEPTR LINEBUF 0) (\MTP.CLRBUF LINEBUF OUTS) (BOUT OUTS (CHARCODE EOL)) (\MTP.INDENT INS OUTS END LMARGIN) (SETQ CARRY (SETQ COLUMN LMARGIN))) (BOUT LINEBUF BYTE))

when (IGREATERP COLUMN LIMIT) do [if (SETQ SPACE (for SPACE in SPACES thereis (LEQ SPACE EDGE))) then (* dump line up to space) (COPYBYTES LINEBUF OUTS 0 (SUB1 (IDIFFERENCE SPACE CARRY))) (BIN LINEBUF) (* eat up space) (SETQ COLUMN (IPLUS LMARGIN (IDIFFERENCE COLUMN SPACE))) else (* punt) (COPYBYTES LINEBUF OUTS 0 (IDIFFERENCE RMARGIN CARRY)) (SETQ COLUMN (ADD1 (IPLUS LMARGIN MTP.RIGHTMARGINWIDTH)) (BOUT OUTS (CHARCODE EOL)) (\MTP.INDENT INS OUTS END LMARGIN) (\MTP.CLRBUF LINEBUF OUTS) (SETQ SPACES) (SETQ CARRY COLUMN))

finally (SETFILEPTR LINEBUF 0) (COPYBYTES LINEBUF OUTS)]

(\MTP.INDENT

[LAMBDA (INS OUTS END LMARGIN) (* drc%: "18-May-86 18:31")

(* * indent OUTS to LMARGIN, unless at end of INS or on an empty line)

(if (AND (ILESSP (GETFILEPTR INS) END) (NEQ (PEEKCCODE INS) (CHARCODE EOL))) then (to LMARGIN do (BOUT OUTS (CHARCODE SPACE)))

(\MTP.CLRBUF

[LAMBDA (INS OUTS) (* drc%: "30-Apr-86 00:14")

(* * Flush INS to OUTS, and then clear INS)

(COPYBYTES INS OUTS) (\SETEOFPTR INS 0) (SETFILEPTR INS 0)

(\MTP.PRINTADDRESSES

[LAMBDA (ADDRESSLIST STREAM) (* bvm%: "20-Dec-83 18:20")

(for ADDR in ADDRESSLIST bind NTHTIME when ADDR do (COND (NTHTIME (PRIN1 " , " STREAM)) (T (SETQ NTHTIME T))) (PRIN1 ADDR STREAM))

(TERPRI STREAM))

)

(RPAQ? MTP.SERVER)

(RPAQ? MTP.LINELENGTH 70)

(RPAQ? MTP.RIGHTMARGINWIDTH 10)

(RPAQ? MTP.FILLMSGFLG %' ASK)

(RPAQ? MTP.INSERTANSWERFLG T)

T)
(replace MTPSTATE of MAILBOX with NEXTSTATE])
(ERROR "Mailbox not in good state for NEXTMESSAGE" MAILBOX])

(MTP.RETRIEVEMESSAGE

[LAMBDA (MAILBOX OUTSTREAM) (* bvm%: " 6-JUL-83 14:27")
(SELECTQ (fetch MTPSTATE of MAILBOX)
(MESSAGE [COND
((EQ (FTPGETMARK (fetch MTPIN of MAILBOX))
(MARK# HERE-IS-FILE))
(\FTP.FLUSH.TO.MARK (fetch MTPIN of MAILBOX)
OUTSTREAM)
(replace MTPSTATE of MAILBOX with (\MTP.ENDOFMESSAGESTATE (fetch MTPIN of MAILBOX])
(\FTPERROR])

(MTP.CLOSEMAILBOX

[LAMBDA (MAILBOX FLUSHP) (* bvm%: " 9-May-84 15:35")
(COND
((BSPOENP (fetch MTPIN of MAILBOX))
(PROG1 [COND
((AND FLUSHP (EQ (fetch MTPSTATE of MAILBOX)
'EMPTY))
(FTPPUTMARK (fetch MTPOUT of MAILBOX)
(MARK# FLUSH-MAILBOX))
(.EOC. (fetch MTPOUT of MAILBOX))
(SELECTC (FTPGETMARK (fetch MTPIN of MAILBOX))
((MARK# YES)
(FTPGETCODE (fetch MTPIN of MAILBOX))
(\FTP.FLUSH.TO.EOC (fetch MTPIN of MAILBOX)
(.FTPDEBUGLOG.))
T)
((MARK# NO)
(FTPGETCODE (fetch MTPIN of MAILBOX))
(\FTP.FLUSH.TO.EOC (fetch MTPIN of MAILBOX)
PROMPTWINDOW)
'?')
(PROGN (\FTPERROR)
'?])
(CLOSEBSPSTREAM (fetch MTPIN of MAILBOX)
5000))])

(DEFINEQ

(MTP.ENDOFMESSAGESTATE

[LAMBDA (INSTREAM) (* bvm%: " 5-SEP-83 18:08")
(SELECTC (FTPGETMARK INSTREAM)
((MARK# HERE-IS-PLIST)
'OPEN)
((MARK# YES)
(FTPGETCODE INSTREAM)
(\FTP.FLUSH.TO.EOC INSTREAM (.FTPDEBUGLOG.))
'EMPTY)
((MARK# NO)
(FTPGETCODE INSTREAM)
(\FTP.FLUSH.TO.EOC INSTREAM PROMPTWINDOW)
'ERROR)
(\FTPERROR])

(MTP.POLLNEWMAIL

[LAMBDA (HOSTPORT USER) (* drc%: "25-Apr-86 12:28")
(** Does a Laurel-style mail check for USER on machine HOSTPORT, returning NIL
(timeout) or a cons of the PUP type of the response and the contents of the response)
(LET ((SOC (\GETMISC SOCKET))
(OUTPUP (ALLOCATE.PUP))
INPUP RESPONSE)
(SETUPPUP OUTPUP HOSTPORT \PUP SOCKET.MISC SERVICES \PT.LAURELCHECK NIL SOC T)
(PUTPUPSTRING OUTPUP USER)
[SETQ RESPONSE (to \MAXETHERTRIES when (SETQ INPUP (EXCHANGE PUPS SOC OUTPUP NIL T))
do (RETURN (CONS (fetch PUPTYPE of INPUP)
(GETPUPSTRING INPUP)))
finally (AND PUPTRACEFLG (printout PUPTRACEFILE "Mail check timed out" T]
(AND INPUP (RELEASE.PUP INPUP))
(RELEASE.PUP OUTPUP)
RESPONSE])

(ADDOVAR MAILSERVERTYPES (MTP MTP.POLLNEWMAIL MTP.OPENMAILBOX MTP.NEXTMESSAGE MTP.RETRIEVEMESSAGE
MTP.CLOSEMAILBOX ETHERPORT))

```

{MEDLEY}<obsolete>lispusers>MTP.;1

(FILESLoad LAFITE)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD MTPMAILBOX (MTPIN MTPOUT MTPSTATE))

(RECORD MTPPARSE (FROMLINE MAILBOX EOH DATELINE))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUPSOCKET.MTP 7)

(RPAQQ \PUPSOCKET.MISCSERVICES 4)

(CONSTANTS \PUPSOCKET.MTP \PUPSOCKET.MISCSERVICES)
)

(RPAQQ PUPTYPES
  ((\PT.ECHOME 1)
   (\PT.IAMECHO 2)
   (\PT.IAMBADECHO 3)
   (\PT.ERROR 4)
   (\PT.RFC 8)
   (\PT.ABORT 9)
   (\PT.END 10)
   (\PT.ENDREPLY 11)
   (\PT.DATA 16)
   (\PT.ADATA 17)
   (\PT.ACK 18)
   (\PT.MARK 19)
   (\PT.INTERRUPT 20)
   (\PT.INTERRUPTREPLY 21)
   (\PT.AMARK 22)
   (\PT.GATEWAYREQUEST 128)
   (\PT.GATEWAYRESPONSE 129)
   (\PT.ALTOTIMEREQUEST 134)
   (\PT.ALTOTIMERESPONSE 135)
   (\PT.MSGCHECK 136)
   (\PT.NEWMAIL 137)
   (\PT.NONEWMAIL 138)
   (\PT.NOMAILBOX 139)
   (\PT.LAURELCHECK 140)
   (\PT.NAMELOOKUP 144)
   (\PT.NAMERESPONSE 145)
   (\PT.NAME/ADDRERROR 146)
   (\PT.ADDRLOOKUP 147)
   (\PT.ADDRRESPONSE 148)
   (\PT.PRINTERSTATUS 128)
   (\PT.STATUSRESPONSE 129)
   (\PT.PRINTERCAPABILITY 130)
   (\PT.CAPABILITYRESPONSE 131)
   (\PT.PRINTJOBSTATUS 132)
   (\PT.PRINTJOBRESPONSE 133)
   (\PT.WHEREUSERREQUEST 152)
   (\PT.WHEREUSERRESPONSE 153)
   (\PT.WHEREUSERERROR 154)
   (\PT.AUTHREQ 168)
   (\PT.AUTHPOSRESP 169)
   (\PT.AUTHNEGRESP 170)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \PT.ECHOME 1)

(RPAQQ \PT.IAMECHO 2)

(RPAQQ \PT.IAMBADECHO 3)

(RPAQQ \PT.ERROR 4)

(RPAQQ \PT.RFC 8)

(RPAQQ \PT.ABORT 9)

(RPAQQ \PT.END 10)

(RPAQQ \PT.ENDREPLY 11)

(RPAQQ \PT.DATA 16)

(RPAQQ \PT.ADATA 17)

(RPAQQ \PT.ACK 18)

```

(RPAQQ \PT.MARK 19)
(RPAQQ \PT.INTERRUPT 20)
(RPAQQ \PT.INTERRUPTREPLY 21)
(RPAQQ \PT.AMARK 22)
(RPAQQ \PT.GATEWAYREQUEST 128)
(RPAQQ \PT.GATEWAYRESPONSE 129)
(RPAQQ \PT.ALTOTIMEREQUEST 134)
(RPAQQ \PT.ALTOTIMERRESPONSE 135)
(RPAQQ \PT.MSGCHECK 136)
(RPAQQ \PT.NEWMAIL 137)
(RPAQQ \PT.NONEWMAIL 138)
(RPAQQ \PT.NOMAILBOX 139)
(RPAQQ \PT.LAURELCHECK 140)
(RPAQQ \PT.NAMELOOKUP 144)
(RPAQQ \PT.NAMERESPONSE 145)
(RPAQQ \PT.NAME/ADDRERROR 146)
(RPAQQ \PT.ADDRLOOKUP 147)
(RPAQQ \PT.ADDRRESPONSE 148)
(RPAQQ \PT.PRINTERSTATUS 128)
(RPAQQ \PT.STATUSRESPONSE 129)
(RPAQQ \PT.PRINTERCAPABILITY 130)
(RPAQQ \PT.CAPABILITYRESPONSE 131)
(RPAQQ \PT.PRINTJOBSTATUS 132)
(RPAQQ \PT.PRINTJOBRESPONSE 133)
(RPAQQ \PT.WHEREUSERREQUEST 152)
(RPAQQ \PT.WHEREUSERRESPONSE 153)
(RPAQQ \PT.WHEREUSERERROR 154)
(RPAQQ \PT.AUTHREQ 168)
(RPAQQ \PT.AUTHPOSRESP 169)
(RPAQQ \PT.AUTHNEGRESP 170)

(CONSTANTS (\PT.ECHOME 1)
 (\PT.IAMECHO 2)
 (\PT.IAMBADECHO 3)
 (\PT.ERROR 4)
 (\PT.RFC 8)
 (\PT.ABORT 9)
 (\PT.END 10)
 (\PT.ENDREPLY 11)
 (\PT.DATA 16)
 (\PT.ADATA 17)
 (\PT.ACK 18)
 (\PT.MARK 19)
 (\PT.INTERRUPT 20)
 (\PT.INTERRUPTREPLY 21)
 (\PT.AMARK 22)
 (\PT.GATEWAYREQUEST 128)
 (\PT.GATEWAYRESPONSE 129)
 (\PT.ALTOTIMEREQUEST 134)
 (\PT.ALTOTIMERRESPONSE 135)
 (\PT.MSGCHECK 136)
 (\PT.NEWMAIL 137)
 (\PT.NONEWMAIL 138)
 (\PT.NOMAILBOX 139)
 (\PT.LAURELCHECK 140)
 (\PT.NAMELOOKUP 144)
 (\PT.NAMERESPONSE 145)
 (\PT.NAME/ADDRERROR 146)


```
(\PT.ADDRLOOKUP 147)
(\PT.ADDRRESPONSE 148)
(\PT.PRINTERSTATUS 128)
(\PT.STATUSRESPONSE 129)
(\PT.PRINTERCAPABILITY 130)
(\PT.CAPABILITYRESPONSE 131)
(\PT.PRINTJOBSTATUS 132)
(\PT.PRINTJOBRESPONSE 133)
(\PT.WHEREUSERREQUEST 152)
(\PT.WHEREUSERRESPONSE 153)
(\PT.WHEREUSERERROR 154)
(\PT.AUTHREQ 168)
(\PT.AUTHPOSRESP 169)
(\PT.AUTHNEGRESP 170))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MTP.SERVER MTP.LINELENGTH MTP.RIGHTMARGINWIDTH MTP.FILLMSGFLG MTP.INSERTANSWERFLG
MTP.INSERTANSWERNSPACES \LAPARSE.FULL LAFITEEDITORFONT UNSUPPLIEDFIELDSTR MESSAGESTR \LAFITEUSERDATA
MAILSERVERTYPES \LAFITE.AUTHENTICATION.FAILURE)
)

(FILESLoad (LOADCOMP)
LAFITE DPUPFTP)
)

(PUTPROPS MTP COPYRIGHT ("Xerox Corporation" 1983 1984 1986))
```

FUNCTION INDEX

MTP.CLOSEMAILBOX	6	MTP.OPENMAILBOX	5	\MTP.CLRBUF	4	\MTP.POLLNEWMAIL	6
MTP.DELIVERMESSAGE	1	MTP.POLLNEWMAIL	5	\MTP.COERCE.MSG	3	\MTP.PRINTADDRESSES	4
MTP.GET.USERDATA	1	MTP.PREPARE.SEND	2	\MTP.ENDOFMESSAGESTATE ..	6		
MTP.MAKEANSWERFORM	2	MTP.RETRIEVEMESSAGE	6	\MTP.FILL	4		
MTP.NEXTMESSAGE	5	\MTP.AUTHENTICATE	3	\MTP.INDENT	4		

CONSTANT INDEX

\PT.ABORT	8	\PT.CAPABILITYRESPONSE ..	8	\PT.INTERRUPTREPLY	8	\PT.PRINTERSTATUS	8
\PT.ACK	8	\PT.DATA	8	\PT.LAURELCHECK	8	\PT.PRINTJOBRESPONSE	8
\PT.ADATA	8	\PT.ECHOME	8	\PT.MARK	8	\PT.PRINTJOBSTATUS	8
\PT.ADDRLOOKUP	8	\PT.END	8	\PT.MSGCHECK	8	\PT.RFC	8
\PT.ADDRRESPONSE	8	\PT.ENDREPLY	8	\PT.NAME/ADDRERROR	8	\PT.STATUSRESPONSE	8
\PT.ALTOTIMEREQUEST	8	\PT.ERROR	8	\PT.NAMELOOKUP	8	\PT.WHEREUSERERROR	8
\PT.ALTOTIMERRESPONSE	8	\PT.GATEWAYREQUEST	8	\PT.NAMERESPONSE	8	\PT.WHEREUSERREQUEST	8
\PT.AMARK	8	\PT.GATEWAYRESPONSE	8	\PT.NEWMAIL	8	\PT.WHEREUSERRESPONSE	8
\PT.AUTHNEGRESP	8	\PT.IAMBADECHO	8	\PT.NOMAILBOX	8	\PUPSOCKET.MISCSERVICES ..	7
\PT.AUTHPOSRESP	8	\PT.IAMECHO	8	\PT.NONEWMAIL	8	\PUPSOCKET.MTP	7
\PT.AUTHREQ	8	\PT.INTERRUPT	8	\PT.PRINTERCAPABILITY	8		

VARIABLE INDEX

LAFITEMODELST	3	MTP.INSERTANSWERFLG	4	MTP.RIGHTMARGINWIDTH	4
MAILSERVERTYPES	6	MTP.INSERTANSWERNSPACES ..	5	MTP.SERVER	4
MTP.FILLMSGFLG	4	MTP.LINELENGTH	4	PUPTYPES	7

RECORD INDEX

MTPMAILBOX	7	MTPPARSE	7
------------------	---	----------------	---
