

File created: 16-May-90 16:10:35 {DSK}<usr>local>lde>lispcore>sources>DPUPFTP.;2

changes to: (VARS DPUPFTPCOMS)

previous date: 19-Aug-88 12:45:25 {DSK}<usr>local>lde>lispcore>sources>DPUPFTP.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

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

(RPAQQ DPUPFTPCOMS

[

::: Implementation of the PUP FTP device

```
(COMS (FNS \FTPINIT \FTPEVENTFN \FTP.OPENFILE \FTP.REOPENFILE \FTP.OPENFILE.FROM.PLIST \FTP.GETFILENAME
\FTP.RECOGNIZEFILE \FTP.DIRECTORYNAMEP \FTP.CLOSEFILE \FTP.REGISTER \FTP.UNREGISTER
\FTP.RENAMEFILE \FTP.DELETEFILE \FTP.GENERATEFILES \FTP.NEXTFILE \FTP.FILEINFOFN
\FTP.GETFILEINFO \FTP.GETFILEINFO.FROM.PROPS \FTP.FROM.LISP.ATTRIBUTE)
(INITVARS (\FTPAVAILABLE)
(\FTP.IDLE.TIMEOUT 120000)
(*FTP-IGNORE-SERVER-FULL*)))
(COMS ;; internal
(FNS \FTP.OPEN.CONNECTION FTP.BREAKCONNECTION \FTP.SENDVERSION \FTP.WHENCLOSED \GETFTPCONNECTION
\RELEASE.FTPCONNECTION \FTP.ERRORHANDLER \FTP.FIX.BROKEN.INPUT \FTP.CLEANUP
\FTP.ASSURE.CLEANUP)
(ADDVARS (\FTPCONNECTIONS))
(FNS \FTP.HANDLE.NO \FTP.DIRECTORYNAMEONLY \FTP.EOL.FROM.PLIST \FTP.MAKELIST \FTP.PRINTPLIST
\FTP.PACKFILENAME \FTP.ADD.QUOTES \FTP.PACK.DIRECTORYNAMEP \FTP.UNPACKFILENAME
\FTP.ADD.USERINFO \FTP.FLUSH.TO.EOC \FTP.FLUSH.TO.MARK \FTPERROR))
(COMS ;; for debugging
(FNS FTPDEBUG FTPPRINTMARK FTPPRINTCODE FTPGETMARK FTPPUTMARK FTPPUTCODE FTPGETCODE)
(INITVARS (FTPDEBUGLOG)
(FTPDEBUGFLG)))
(DECLARE%: EVAL@COMPILE DONTCOPY (VARS FTPMARKTYPES)
(CONSTANTS \FTP.VERSION)
(CONSTANTS * FTPNOCODES)
(MACROS MARK# .EOC. .FTPDEBUGLOG.)
(PROP INFO MARK#)
(RECORDS FTPCONNECTION FTPSTREAM FTPFILEGENSTATE)
(GLOBALVARS FTPDEBUGFLG \FTPCONNECTIONS \FTPAVAILABLE \FTP.IDLE.TIMEOUT \BSPFDEV \FTPDEV))
(DECLARE%: DONTEVAL@LOAD DOCOPY (P (\FTPINIT]))
```

::: Implementation of the PUP FTP device

(DEFINEQ
(\FTPINIT

[LAMBDA NIL

(COND

((type? FDEV \BSPFDEV)

(SETQ \FTPDEV (NCREATE 'FDEV \BSPFDEV))

(with FDEV \FTPDEV (SETQ DEVICENAME 'DPUPFTP)

(SETQ OPENFILE (FUNCTION \FTP.OPENFILE))

(SETQ REOPENFILE (FUNCTION \FTP.REOPENFILE))

(SETQ CLOSEFILE (FUNCTION \FTP.CLOSEFILE))

(SETQ DIRECTORYNAMEP (FUNCTION \FTP.DIRECTORYNAMEP))

(SETQ GETFILENAME (FUNCTION \FTP.GETFILENAME))

(SETQ GETFILEINFO (FUNCTION \FTP.GETFILEINFO))

(SETQ RENAMEFILE (FUNCTION \FTP.RENAMEFILE))

(SETQ DELETEFILE (FUNCTION \FTP.DELETEFILE))

(SETQ GENERATEFILES (FUNCTION \FTP.GENERATEFILES))

(SETQ EVENTFN (FUNCTION \FTPEVENTFN))

(SETQ OPENP (FUNCTION \GENERIC.OPENP))

(SETQ REGISTERFILE (FUNCTION \FTP.REGISTER))

(SETQ UNREGISTERFILE (FUNCTION \FTP.UNREGISTER)))

(SETQ \FTPAVAILABLE T])

; Edited 17-Nov-87 14:43 by bvm:

; Specialize the BSP device

(\FTPEVENTFN

[LAMBDA (DEV EVENT)

(SELECTQ EVENT

(BEFORELOGOUT (FTP.BREAKCONNECTION T))

((BEFORESYSOUT BEFOREMAKESYS BEFORESAVEVM)

(FTP.BREAKCONNECTION T T))

NIL)

(\BSPEVENTFN DEV EVENT])

(* bvm%: "28-Apr-85 14:32")

(\FTP.OPENFILE

; Edited 16-Nov-87 17:17 by bvm

```

[LAMBDA (FILENAME ACCESS RECOG OTHERINFO)
  (RESETLST
    (PROG (HOST DESIREDPLIST TYPE BYTESIZE EOLCONVENTION)
      (COND
        ((SELECTQ ACCESS
          (INPUT (EQ RECOG 'NEW))
          (OUTPUT (EQ RECOG 'OLD))
          T)
          (LISPERROR "FILE WON'T OPEN" FILENAME)))
        (OR (SETQ HOST (\FTP.UNPACKFILENAME FILENAME))
          (RETURN))
        (SETQ DESIREDPLIST (CDR HOST))
        (SETQ HOST (CAR HOST))
        (SELECTQ ACCESS
          (OUTPUT [for PAIR in OTHERINFO when (LISTP PAIR)
            do (COND
              ((SELECTQ (CAR PAIR)
                ((TYPE FILETYPE)
                  [SELECTQ (SETQ TYPE (CADR PAIR))
                    (TEXT T)
                    (NIL)
                    (PROGN ; All unrecognized types are BINARY
                      (SETQ TYPE 'BINARY)
                      NIL)
                    (BYTESIZE (SETQ BYTESIZE (OR (FIXP (CADR PAIR))
                      (LISPERROR "ILLEGAL ARG" PAIR)))
                      NIL)
                    ((EOL EOLCONVENTION)
                      (SETQ EOLCONVENTION (CADR PAIR))
                      NIL)
                    ((CREATIONDATE ICREATIONDATE)
                      [push DESIREDPLIST (LIST 'CREATION-DATE
                        (if (EQ (CAR PAIR)
                          'ICREATIONDATE)
                          then (GDATE (CADR PAIR)
                            (DATEFORMAT TIME.ZONE))
                          else (CADR PAIR])
                        NIL)
                      (LENGTH [push DESIREDPLIST (LIST 'SIZE (OR (FIXP (CADR PAIR))
                        (LISPERROR "ILLEGAL ARG"
                          PAIR])
                        NIL)
                      ((SEQUENTIAL DON'TCACHE)
                        NIL)
                      T)
                      (push DESIREDPLIST PAIR]
                    [push DESIREDPLIST (LIST 'TYPE (OR TYPE (SETQ TYPE DEFAULTFILETYPE)
                      (SELECTQ TYPE
                        (TEXT [push DESIREDPLIST (LIST 'END-OF-LINE-CONVENTION (OR EOLCONVENTION
                          'CR])
                        (BINARY (push DESIREDPLIST (LIST 'BYTE-SIZE (OR BYTESIZE 8))))
                        NIL))
                      NIL)
                    (RETURN (\FTP.OPENFILE.FROM.PLIST HOST DESIREDPLIST ACCESS))))))

```

(\FTP.REOPENFILE

; Edited 17-Nov-87 14:43 by bvm:

```

[LAMBDA (NAME ACCESS RECOG OTHERINFO FDEV STREAM)
  ;; This is for the cruffy REVALIDATEFILELST that the Leaf device (where all our open files are registered) does after logout, etc. Simplest to just
  ;; pass the file back as though all is ok; we will actually reopen it in the broken connection error handler when somebody tries to read more from the
  ;; stream. Of course, if file was open for output, all is lost.
  (AND (EQ ACCESS 'INPUT)
    STREAM])

```

(\FTP.OPENFILE.FROM.PLIST

(* bvm%: "28-Apr-85 14:14")

```

[LAMBDA (HOST DESIREDPLIST ACCESS)
  (PROG (CONNECTION INS OUTS REMOTEPLIST FULLNAME)
    NEWCONNECTION
    (OR (SETQ CONNECTION (\GETFTPCONNECTION HOST T))
      (RETURN))
    (SETQ INS (fetch FTPIN of CONNECTION))
    (SETQ OUTS (fetch FTPOUT of CONNECTION))
  RETRY
    (FTPPUTMARK OUTS (SELECTQ ACCESS
      (INPUT (MARK# RETRIEVE))
      (OUTPUT (MARK# NEW-STORE))
      NIL))
    (\FTP.PRINTPLIST OUTS DESIREDPLIST)
    (.EOC. OUTS)
    (SELECTC (FTPGETMARK INS)
      ((MARK# NO)
        (COND
          [(\FTP.HANDLE.NO CONNECTION DESIREDPLIST)

```

```

(COND
  ((BSOPENP INS 'INPUT)
   (GO RETRY))
  (T (GO NEWCONNECTION)
   (T (\RELEASE.FTPCONNECTION CONNECTION)
    (RETURN))))
(MARK# HERE-IS-PLIST)
(SETQ REMOTEPLIST (READPLIST INS))
[SETQ FULLNAME (\FTP.PACKFILENAME HOST REMOTEPLIST NIL (CADR (ASSOC 'DEVICE DESIREDPLIST)
 (OR (EQ (\FTPGETMARK INS)
         (MARK# EOC))
      (RETURN (\FTPERROR CONNECTION))))))
(MARK# BROKEN)
(GO NEWCONNECTION))
(RETURN (\FTPERROR CONNECTION))
(SELECTQ ACCESS
 (INPUT (\FTPPUTMARK OUTS (MARK# YES))
        (\FTPPUTCODE OUTS 0)
        (.EOC. OUTS)
        (SELECTC (\FTPGETMARK INS)
                 (MARK# NO)
                 [COND
                  ((\FTP.HANDLE.NO CONNECTION DESIREDPLIST NIL NIL T)
                   (CLOSEBSPSTREAM INS 2000) ; Can't recover from in the middle like this, so just flush and start
                                               ; over
                   (GO NEWCONNECTION))
                  (T (\RELEASE.FTPCONNECTION CONNECTION)
                   (RETURN (LISPERROR "FILE WON'T OPEN" FULLNAME]))
                  ((MARK# HERE-IS-FILE)
                   (replace FULLFILENAME of INS with FULLNAME)
                   (replace FTPFILEPROPS of INS with REMOTEPLIST)
                   (replace ACCESS of INS with 'INPUT)
                   (replace EOLCONVENTION of INS with (\FTP.EOL.FROM.PLIST REMOTEPLIST))
                   (\BSP.DECLARE.FILEPTR INS 0) ; For GETFILEPTR
                   (replace DEVICE of INS with \FTPFDEV)
                   (RETURN INS))
                  ((MARK# BROKEN)
                   (GO NEWCONNECTION))
                  (RETURN (\FTPERROR CONNECTION))))))
(OUTPUT (COND
         ((BSOPENP OUTS 'OUTPUT)
          (\FTPPUTMARK OUTS (MARK# HERE-IS-FILE))
          (replace FULLFILENAME of OUTS with FULLNAME)
          (replace FTPFILEPROPS of OUTS with REMOTEPLIST)
          (\BSP.DECLARE.FILEPTR OUTS 0)
          (replace EOLCONVENTION of OUTS with (\FTP.EOL.FROM.PLIST DESIREDPLIST))
          (replace DEVICE of OUTS with \FTPFDEV)
          (RETURN OUTS))
         (T (GO NEWCONNECTION))))
NIL])

```

(\FTP.GETFILENAME

```

[LAMBDA (NAME RECOG DEV)
 (SELECTQ RECOG
  ((OLD OLDEST)
   (\FTP.RECOGNIZEFILE NAME DEV NIL RECOG))
  ((OLD/NEW NEW)
   (OR (\FTP.RECOGNIZEFILE NAME DEV NIL RECOG)
        (PACKFILENAME 'BODY NAME 'VERSION 1)))
  (SHOULDNT])

```

; Edited 16-Nov-87 18:52 by bvm

(\FTP.RECOGNIZEFILE

```

[LAMBDA (NAME DEV OPTION RECOG DESIREDPROPS)
 (RESETLST
  (PROG (CONNECTION HOST INS OUTS REMOTEPLIST DESIREDPLIST BESTVERSION V BESTPLIST RESULT CODE
        DEVICEWANTED)
  (OR (SETQ HOST (\FTP.UNPACKFILENAME NAME))
      (RETURN))
  (SETQ DESIREDPLIST (CDR HOST))
  (SETQ HOST (CAR HOST))
  [if (EQ OPTION 'DIRECTORYNAMEP)
    then
      (RPLACA (CDR (ASSOC 'NAME-BODY DESIREDPLIST))
              "QXZYQJ")
    else (push DESIREDPLIST ' (DESIRED-PROPERTY VERSION))
  [if (NULL (ASSOC 'VERSION DESIREDPLIST))
    then
      (if (SETQ V (SELECTQ (GETHOSTINFO HOST 'OSTYPE)
                          (VMS (if (EQ RECOG 'OLDEST)
                                   then "-0"
                                   else "0"))
                          ((TENEX TOPS20)
                           (if (EQ RECOG 'OLDEST)
                               then "-2"

```

; Edited 19-Aug-88 12:04 by bvm

; Give random fake name

; Will need VERSION attribute to discriminate versions

; Try to restrict enumeration to a single version for speed

```

        (if (EQ RECOG 'OLDEST)
            then "-2"

```

```

                else "0")
                ((NIL IFS D)
                 (if (EQ RECOG 'OLDEST)
                     then "L"
                     else "H"))
                NIL))
            then (push DESIREDPLIST (LIST 'VERSION V)
                 (if (NEQ OPTION 'PROPS)
                     then (push DESIREDPLIST '(DESIRED-PROPERTY NAME-BODY)
                          ; PROPS doesn't care about the actual name
                          (push DESIREDPLIST '(DESIRED-PROPERTY DIRECTORY)
                           (if (SETQ DEVICEWANTED (CADR (ASSOC 'DEVICE DESIREDPLIST)))
                               then (push DESIREDPLIST '(DESIRED-PROPERTY DEVICE)
                                    [for PROP in DESIREDPROPS do (push DESIREDPLIST '(DESIRED-PROPERTY ,PROP)
                                         NEWCONNECTION
                                         (OR (SETQ CONNECTION (\GETFTPCONNECTION HOST T T))
                                             (RETURN))
                                         (SETQ INS (fetch FTPIN of CONNECTION))
                                         (SETQ OUTS (fetch FTPOUT of CONNECTION))
                                         RETRY
                                         (FTPPUTMARK OUTS (MARK# ENUMERATE))
                                         (\FTP.PRINTPLIST OUTS DESIREDPLIST)
                                         (.EOC. OUTS)
                                         LP (SELECTC (FTPGETMARK INS)
                                             ((MARK# NO)
                                              [SELECTC (SETQ CODE (FTPGETCODE INS T))
                                                  (\NO.ILLEGAL.DIRECTORY
                                                   (\FTP.FLUSH.TO.EOC INS (.FTPDEBUGLOG.))
                                                  (\NO.FILE.NOT.FOUND
                                                   (\FTP.FLUSH.TO.EOC INS (.FTPDEBUGLOG.))
                                                  [COND
                                                    (EQ OPTION 'DIRECTORYNAMEP) ; Directory exists
                                                    (SETQ RESULT (\FTP.PACK.DIRECTORYNAMEP CONNECTION DESIREDPLIST])
                                                    (COND
                                                      ((\FTP.HANDLE.NO CONNECTION DESIREDPLIST NIL CODE NIL T)
                                                       (COND
                                                         ((BSOPENP INS 'INPUT)
                                                          (GO RETRY))
                                                         (T (GO NEWCONNECTION])
                                                       (MARK# HERE-IS-PLIST) ; Have to remember the plist corresponding to the best version.
                                                       (SETQ REMOTEPLIST (READPLIST INS))
                                                       (if [OR [NULL (SETQ V (CADR (ASSOC 'VERSION REMOTEPLIST)
                                                           (PROGN (SETQ V (MKATOM V))
                                                                (OR (NULL BESTVERSION)
                                                                    (SELECTQ RECOG
                                                                      (OLDEST (< V BESTVERSION))
                                                                      (> V BESTVERSION))
                                                                    then (SETQ BESTVERSION V)
                                                                    (SETQ BESTPLIST REMOTEPLIST))
                                                                (GO LP))
                                                            (MARK# BROKEN)
                                                            (GO NEWCONNECTION))
                                                            (MARK# EOC)
                                                            [SETQ RESULT (SELECTQ OPTION
                                                                (PROPS BESTPLIST)
                                                                (DIRECTORYNAMEP
                                                                 (\FTP.PACK.DIRECTORYNAMEP CONNECTION BESTPLIST))
                                                                (PROGN (if (AND (EQ RECOG 'NEW)
                                                                    BESTVERSION)
                                                                    then ; For RECOG NEW bump the version
                                                                    (RPLACA (CDR (ASSOC 'VERSION BESTPLIST))
                                                                      (+ BESTVERSION 1)))
                                                                    (\FTP.PACKFILENAME HOST BESTPLIST NIL DEVICEWANTED])
                                                                (\FTPERROR CONNECTION)
                                                                (\RELEASE.FTPCONNECTION CONNECTION)
                                                                (RETURN RESULT)))]))

```

```

(\FTP.DIRECTORYNAMEP
 [LAMBDA (HOST/DIR DEV) ; (* bvm%: "27-SEP-83 17:59")
  (\FTP.RECOGNIZEFILE HOST/DIR DEV 'DIRECTORYNAMEP])

```

```

(\FTP.CLOSEFILE
 [LAMBDA (STREAM) ; Edited 30-Nov-87 16:35 by bvm:
  (PROG ((ACCESS (fetch ACCESS of STREAM))
         [CONN (find C in \FTP.CONNECTIONS suchthat (OR (EQ (fetch FTPIN of C)
                                                             STREAM)
                                                         (EQ (fetch FTPOUT of C)
                                                             STREAM]
         (FILENAME (fetch FULLFILENAME of STREAM))
         INS SUCCESS)
        (replace FTPFILEPROPS of STREAM with NIL)
        (SELECTQ ACCESS
         (INPUT [COND
          ((NOT (BSOPENP STREAM ACCESS)) ; connection went away

```

```

NIL)
([OR (\EOF STREAM)
  (PROGN (\BSP.FLUSHINPUT STREAM)
    (AND (BSOPENP STREAM ACCESS)
      (\EOF STREAM)

;; Hack. We are at the end of the file, or the remainder of the file has been sent, so we can terminate the RETRIEVE
;; cleanly
  (SETQ SUCCESS (SELECTC (FTPGETMARK STREAM)
    ((MARK# YES) ; File sent ok
      (FTPGETCODE STREAM)
      (\FTP.FLUSH.TO.EOC STREAM (.FTPDEBUGLOG.)))
    ((MARK# NO)
      (FTPGETCODE STREAM T)
      (PROG1 (\FTP.FLUSH.TO.EOC STREAM (\GETSTREAM PROMPTWINDOW
        'OUTPUT))
        (ERROR "CLOSEF: Remote file not successfully retrieved")
        ))
    NIL]))
  (OUTPUT (OR (SELECTC (COND
    ((SETQ INS (BSPINPUTSTREAM STREAM))
      (FTPPUTMARK STREAM (MARK# YES))
      (FTPPUTCODE STREAM 0)
      (.EOC. STREAM)
      (FTPGETMARK INS)))
    ((MARK# YES)
      (FTPGETCODE INS)
      (SETQ SUCCESS (\FTP.FLUSH.TO.EOC INS (.FTPDEBUGLOG.)))
    ((MARK# NO)
      (SELECTC (PROG1 (FTPGETCODE INS T)
        (CL:FORMAT PROMPTWINDOW "~&~A: " (fetch FTPHOST of CONN))
        [SETQ SUCCESS (\FTP.FLUSH.TO.EOC INS (\GETSTREAM PROMPTWINDOW
          'OUTPUT])
        (\NO.STORAGE.FULL
          (LISPERROR "FILE SYSTEM RESOURCES EXCEEDED" FILENAME))
        NIL))
      NIL)
    (ERROR "CLOSEF: Remote file not successfully stored" FILENAME)))
  NIL)
(COND
  (SUCCESS (replace DEVICE of STREAM with \BSPFDEV) ; Stream still in good protocol state
    (\RELEASE.FTPCONNECTION CONN)) ; Make it back into a plain BSP stream
  (CONN (CLOSEBSPSTREAM (fetch FTPIN of CONN)
    1000))
  (RETURN FILENAME]))

```

```

(\FTP.REGISTER
[LAMBDA (DEVICE STREAM) (* PAVEL "14-Oct-86 19:09")
  (\ADD-OPEN-STREAM (\GETDEVICEFROMNAME (fetch (STREAM FULLFILENAME) of STREAM)
    STREAM))

```

```

(\FTP.UNREGISTER
[LAMBDA (DEVICE STREAM) (* hdj "1-Oct-86 18:23")
  (\GENERIC-UNREGISTER-STREAM (if (FMEMB STREAM (\DEVICE-OPEN-STREAMS DEVICE))
    then DEVICE
    else (\GETDEVICEFROMNAME (fetch (STREAM FULLNAME) of STREAM)))
  STREAM))

```

```

(\FTP.RENAMEFILE
[LAMBDA (OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE) ; Edited 16-Nov-87 19:02 by bvm
  (COND
    ((NEQ OLD-DEVICE NEW-DEVICE)
      (\GENERIC.RENAMEFILE OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE))
    (T (RESETLST
      [PROG ((HOST (\FTP.UNPACKFILENAME OLDFILE))
        CONNECTION HOST OLDNAME INS OUTS OLDPLIST NEWPLIST CODE)
        (if NIL
          then ; This is junk
            (OR HOST (FDEVOP 'OPENP OLD-DEVICE (FULLNAME OLDFILE)
              NIL OLD-DEVICE)
              (RETURN)))
          (SETQ OLDPLIST (CDR HOST))
          (SETQ HOST (CAR HOST))
          (OR (SETQ NEWPLIST (\FTP.UNPACKFILENAME NEWFILE T))
            (RETURN))
          (if (NOT (STRING-EQUAL (CAR NEWPLIST)
            HOST))
            then ; Different hosts--can't do it. This happens only if both hosts are
              ; ftp-only hosts, in which case they got mapped to the same
              ; device
              (RETURN (\GENERIC.RENAMEFILE OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE))
            else (SETQ NEWPLIST (CDR NEWPLIST)))
          (CLEAR.LEAF.CACHE HOST) ; In case Leaf has this file open for input

```

```

NEWCONNECTION
  (OR (SETQ CONNECTION (\GETFTPCONNECTION HOST T T))
      (RETURN))
  (SETQ INS (fetch FTPIN of CONNECTION))
  (SETQ OUTS (fetch FTPOUT of CONNECTION))
RETRY
  (FTPPUTMARK OUTS (MARK# RENAME))
  (\FTP.PRINTPLIST OUTS OLDPLIST)
  (\FTP.PRINTPLIST OUTS NEWPLIST)
  (.EOC. OUTS)
  (RETURN (if (PROG1 (SELECTC (FTPGETMARK INS)
                            (MARK# NO)
                            [SELECTC (SETQ CODE (FTPGETCODE INS T))
                                      (\NO.UNIMPLEMENTED ;; Concession to stupid DEI
                                      (\FTP.FLUSH.TO.EOC INS (.FTPDEBUGLOG.))
                                      (RETURN (\GENERIC.RENAMEFILE OLD-DEVICE OLDFILE
                                                NEW-DEVICE NEWFILE)))
                            (COND
                              ((\FTP.HANDLE.NO CONNECTION OLDPLIST NIL CODE NIL NIL
                                NEWPLIST)
                               (COND
                                ((BSPOPNP INS 'INPUT)
                                 (GO RETRY))
                                (T (GO NEWCONNECTION])
                              (MARK# YES)
                              (FTPGETCODE INS)
                              (AND (\FTP.FLUSH.TO.EOC INS (.FTPDEBUGLOG.)
                                NEWFILE))
                              (MARK# BROKEN)
                              (GO NEWCONNECTION))
                              (\FTPERROR CONNECTION))
                              (\RELEASE.FTPCONNECTION CONNECTION))
                                then NEWFILE
                                else (LISPERROR "FILE WON'T OPEN" NEWFILE)]))

```

(\FTP.DELETEFILE

```

[LAMBDA (FILENAME) ; Edited 30-Nov-87 16:16 by bvm:
  (RESETLST
   [PROG (CONNECTION HOST REMOTENAME INS OUTS REMOTEPLIST DESIREDPLIST RESULT)
         (OR (SETQ HOST (\FTP.UNPACKFILENAME FILENAME))
             (if NIL
                 then ; This is junk
                     (LET* ((NAME (FULLNAME FILENAME))
                            (DEVICE (\GETDEVICEFROMNAME NAME)))
                       (FDEVOP 'OPENP DEVICE NAME NIL DEVICE)))
                 (RETURN))
         (SETQ DESIREDPLIST (CDR HOST))
         (SETQ HOST (CAR HOST))
         [COND
          ((AND (NULL (ASSOC 'VERSION DESIREDPLIST))
                (EQ (GETHOSTINFO HOST 'OSTYPE)
                    'VMS))
           ; Ugh blech, VMS defaults version to newest, have to explicitly
           ; ask for oldest
           (push DESIREDPLIST (LIST 'VERSION "-0"))
           (for PROP in ' (DIRECTORY NAME-BODY VERSION) do (push DESIREDPLIST (LIST 'DESIRED-PROPERTY PROP)))
           (CLEAR.LEAF.CACHE HOST) ; In case Leaf has this file open for input
          NEWCONNECTION
          (OR (SETQ CONNECTION (\GETFTPCONNECTION HOST T T))
              (RETURN))
          (SETQ INS (fetch FTPIN of CONNECTION))
          (SETQ OUTS (fetch FTPOUT of CONNECTION))
          RETRY
          (FTPPUTMARK OUTS (MARK# DELETE))
          (\FTP.PRINTPLIST OUTS DESIREDPLIST)
          (.EOC. OUTS)
          (SELECTC (FTPGETMARK INS)
                  (MARK# NO)
                  (COND
                   [(\FTP.HANDLE.NO CONNECTION DESIREDPLIST NIL NIL NIL T)
                    (COND
                     ((BSPOPNP INS 'INPUT)
                      (GO RETRY))
                     (T (GO NEWCONNECTION])
                   (MARK# YES)
                   (\RELEASE.FTPCONNECTION CONNECTION)
                   (RETURN NIL))))
                  (MARK# HERE-IS-PLIST)
                  NIL)
                  (MARK# BROKEN)
                  (GO NEWCONNECTION))
                  (RETURN (\FTPERROR CONNECTION)))
          NEXTPLIST
          (SETQ REMOTEPLIST (READPLIST INS))
          (OR (EQ (FTPGETMARK INS)

```

```

(MARK# EOC))
(\FTPERROR CONNECTION))
(FTPPUTMARK OUTS (MARK# YES))
(FTPPUTCODE OUTS 0)
(.EOC. OUTS)
(SELECTC (FTPGETMARK INS)
  ((MARK# NO)
    (COND
      ((\FTP.HANDLE.NO CONNECTION DESIREDPLIST NIL NIL T T)
        (CLOSEBSPSTREAM INS 2000)
        (GO NEWCONNECTION))))
    (MARK# YES)
      (FTPGETCODE INS)
      (\FTP.FLUSH.TO.MARK INS)
      [push RESULT (\FTP.PACKFILENAME HOST REMOTEPLIST NIL (CADR (ASSOC 'DEVICE DESIREDPLIST]))
        (RETURN (\FTPERROR CONNECTION)))

```

;; Got Yes/No on that file, see if there will be more. Usually there is just one file, but this code is prepared to do DELETE *.*....

```

(SELECTC (FTPGETMARK INS)
  ((MARK# HERE-IS-PLIST)
    (GO NEXTPLIST))
  (MARK# EOC)
    (\RELEASE.FTPCONNECTION CONNECTION)
  (RETURN (COND
    ((CDR RESULT)
      (REVERSE RESULT))
    (T (CAR RESULT]))
    (RETURN (\FTPERROR CONNECTION]))))

```

(\FTP.GENERATEFILES

; Edited 19-Aug-88 11:54 by bvm

```

[LAMBDA (DEVICE PATTERN DESIREDPROPS OPTIONS)
  (PROG [(RESULT (RESETLST
    [PROG (CONNECTION HOST REMOTENAME INS OUTS DESIREDPLIST CODE VERSION EXTENSION DEVICE
      WANTDEVICE NAME DIRECTORY NAMEBODY OSTYPE INFO FILTERNEEDED)
      (for TAIL on (UNPACKFILENAME.STRING PATTERN) by (CDDR TAIL)
        do (SELECTQ (CAR TAIL)
          (HOST [SETQ HOST (\CANONICAL.HOSTNAME (MKATOM (CADR TAIL))]
            (DIRECTORY (SETQ DIRECTORY (CADR TAIL)))
            (NAME (SETQ NAME (CADR TAIL)))
            (EXTENSION (SETQ EXTENSION (OR (CADR TAIL)
              "")))
            (VERSION [SETQ VERSION (AND (IGREATERP (NCHARS (CADR TAIL))
              0)
              (MKATOM (CADR TAIL))]
            (DEVICE (SETQ WANTDEVICE (SETQ DEVICE (CADR TAIL))))
            (RETURN)))
          (SETQ OSTYPE (GETHOSTINFO HOST 'OSTYPE))
          (SELECTQ OSTYPE
            (TENEX [COND
              ((AND (STRPOS '* NAME)
                (IGREATERP (NCHARS NAME)
                  1))
                (SETQ FILTERNEEDED (SETQ NAME '*))
              [COND
                (EXTENSION (SELECTQ (NCHARS EXTENSION)
                  (0 ;Maxc enumerates 'name.*' even when given just 'name.'
                    (SETQ FILTERNEEDED T))
                  (1 ;Extension * no problem
                    )
                  (COND
                    ((STRPOS '* EXTENSION)
                      (SETQ FILTERNEEDED (SETQ EXTENSION '*))
                    [OR VERSION (COND
                      ((EQ OSTYPE 'TENEX)
                        (SETQ VERSION 0))
                      (TOPS20 ; Can handle all *s
                        (OR VERSION (SETQ VERSION 0))
                        (OR WANTDEVICE (SETQ WANTDEVICE T)))
                      (VMS ; Can handle all *s
                        )
                      ((NIL IFS UNIX)
                        (COND
                          (EXTENSION (SELECTQ (NCHARS EXTENSION)
                            (1 [COND
                              ((EQ (CHCON1 EXTENSION)
                                (CHARCODE *)))
                              ; If enumerating FOO.* need to ask for FOO* or else we will miss
                              ; extensionless FOO
                              (SETQ EXTENSION NIL)
                              (COND
                                ((NEQ (NTHCHARCODE NAME -1)
                                  (CHARCODE *)))
                                (SETQ FILTERNEEDED (SETQ NAME
                                  (CONCAT NAME
                                    '*]))

```

```

;; Explicit null extension. IFS enumerates FOO. okay, but FOO*. would also
;; enumerate files with non-null extensions
      (SETQ EXTENSION NIL)
      (SETQ FILTERNEEDED (STRPOS '* NAME)))
      NIL)))
[COND
  [(EQ OSTYPE 'UNIX) ; Coerce directory name to lowercase, get rid of trailing /
    [COND
      ((EQ (NTHCHARCODE DIRECTORY -1)
          (CHARCODE /))
        (SETQ DIRECTORY (SUBSTRING DIRECTORY 1 -2))
      [COND
        ((NEQ (NTHCHARCODE DIRECTORY 1)
            (CHARCODE /))
          (SETQ DIRECTORY (CONCAT '/ DIRECTORY])
        (COND
          ((U-CASEP DIRECTORY)
            (SETQ DIRECTORY (L-CASE DIRECTORY])
          (T (OR VERSION (SETQ VERSION 'H])
        NIL)
      (SETQ DESIREDPLIST (for PROP in (NCONC (for PROP in DESIREDPROPS
        collect (\FTP.FROM.LISP.ATTRIBUTE PROP))
        ' (DIRECTORY NAME-BODY VERSION))
        collect (LIST 'DESIRED-PROPERTY PROP)))
    [COND
      (([AND VERSION (OR (NEQ VERSION '*')
        (EQ OSTYPE 'VMS])
        (push DESIREDPLIST (LIST 'VERSION VERSION])
    [SETQ NAMEBODY (COND
      ((NULL EXTENSION)
        NAME)
      (T (CONCAT NAME "." EXTENSION])
    [COND
      ((EQ OSTYPE 'UNIX)
        [COND
          ((AND NIL (U-CASEP NAMEBODY))
            ;; Would like to help out by coercing name to lowercase, but the leaf server really does write
            ;; uppercase filenames!
            (SETQ NAMEBODY (L-CASE NAMEBODY])
          (COND
            ((NEQ (NTHCHARCODE NAMEBODY -1)
                (CHARCODE *))
              ; Unix FTP server does not understand versions, so make sure
              ; that whatever pattern we give ends in *
              (SETQ FILTERNEEDED (SETQ NAMEBODY (CONCAT NAMEBODY '*])
            (push DESIREDPLIST (LIST 'NAME-BODY NAMEBODY))
          [COND
            (DIRECTORY (push DESIREDPLIST (LIST 'DIRECTORY DIRECTORY])
          [COND
            (WANTDEVICE (push DESIREDPLIST (LIST 'DESIRED-PROPERTY 'DEVICE))
              (COND
                (DEVICE (push DESIREDPLIST (LIST 'DEVICE DEVICE])
            (push DESIREDPLIST [LIST 'USER-NAME (CAR (SETQ INFO (\INTERNAL/GETPASSWORD HOST])
              (LIST 'USER-PASSWORD (CDR INFO)))
    NEWCONNECTION
    (OR (SETQ CONNECTION (\GETFTPCONNECTION HOST T T))
      (GO NOFILES))
    (SETQ INS (fetch FTPIN of CONNECTION))
    (SETQ OUTS (fetch FTPOUT of CONNECTION))
  RETRY
  (FTPPUTMARK OUTS (MARK# ENUMERATE))
  (\FTP.PRINTPLIST OUTS DESIREDPLIST)
  (.EOC. OUTS)
  (SELECTC (FTPGETMARK INS)
    (MARK# NO)
    (COND
      [(\FTP.HANDLE.NO CONNECTION DESIREDPLIST NIL CODE NIL T)
        (COND
          ((BSPOPENP INS 'INPUT)
            (GO RETRY))
          (T (GO NEWCONNECTION]
          (T (\RELEASE.FTPCONNECTION CONNECTION))))
    ((MARK# HERE-IS-PLIST)
      (replace FTPBUSY of CONNECTION with (SETUPTIMER \FTP.IDLE.TIMEOUT))
      ; This guy gets a timer because the generator could be aborted
      ; out of our control. Blech
    [RETURN (create FILEGENOBJ
      NEXTFILEFN _ (FUNCTION \FTP.NEXTFILE)
      FILEINFOFN _ (FUNCTION \FTP.FILEINFOFN)
      GENFILESTATE _
      (create FTPFILEGENSTATE
        FTPGENCONNECTION _ CONNECTION
        FTPDEVICEWANTED _ WANTDEVICE
        FTPGENPLIST _ NIL
        FTPNAMEFILTER _ (AND FILTERNEEDED (
          DIRECTORY.MATCH.SETUP

```


PATTERN]]

```

      ((MARK# BROKEN)
       (GO NEWCONNECTION))
      (\FTPERROR CONNECTION))
NOFILES
      (RETURN (create FILEGENOBJ
                NEXTFILEFN _ (FUNCTION NIL)))]
[COND
  ((AND RESULT (fetch GENFILESTATE of RESULT)) ; Have a generator, so need to assure generator will terminate
   (COND
    [(EQMEMB 'RESETLST OPTIONS)
     (RESETSAVE NIL (LIST [FUNCTION (LAMBDA (CONNECTION)
                                (AND RESETSTATE (CLOSEBSPSTREAM (fetch FTPIN of CONNECTION)
                                                                0]
                                (fetch FTPGENCONNECTION of (fetch GENFILESTATE of RESULT]
     (T (\FTP.ASSURE.CLEANUP]
   (RETURN RESULT])

```

(\FTP.NEXTFILE

```

[LAMBDA (GENSTATE NAMEONLY) ; (* bvm%: "13-Jul-84 16:44")
  (DECLARE (SPECVARS FTPCONNECTION) ; Seen by \FTP.CLEANUP
  (PROG ((FTPCONNECTION (fetch FTPGENCONNECTION of GENSTATE))
        (FILTER (fetch FTPNAMEFILTER of GENSTATE))
        INS NAMEBODY NAME EXT N PLIST)
  (COND
    ((NULL FTPCONNECTION)
     (RETURN (ERROR "End of file Enumerator" GENSTATE)))
    ((NOT (SETQ INS (fetch FTPIN of FTPCONNECTION)))
     (GO BROKEN)))
  LP (SETUPTIMER \FTP.IDLE.TIMEOUT (fetch FTPBUSY of FTPCONNECTION))
  [COND
    ((\EOF INS)
     ;; NEW-ENUMERATE sends plists one after another with no intervening HERE-IS-PLIST; check here for oldstyle, or for end of
     ;; command
     (SELECTC (FTPGETMARK INS)
      ((MARK# EOC)
       ((RELEASE.FTPCONNECTION FTPCONNECTION)
        (replace FTPGENCONNECTION of GENSTATE with NIL)
        (RETURN NIL))
       ((MARK# HERE-IS-PLIST) ; Old style
        )
       ((MARK# BROKEN)
        (GO BROKEN))
       (RETURN (\FTPERROR FTPCONNECTION]
    (COND
      ((AND (NULL (SETQ PLIST (READPLIST INS)))
            (NOT (BSOPENP INS 'INPUT)
            (GO BROKEN)))
      [SETQ NAME (COND
        (NAMEONLY (OR (CADR (ASSOC 'NAME-BODY PLIST))
                      ""))
        (T (\FTP.PACKFILENAME (fetch FTPHOST of FTPCONNECTION)
                               PLIST T (fetch FTPDEVICEWANTED of GENSTATE]
    (COND
      ((AND FILTER (NOT (DIRECTORY.MATCH FILTER NAME)))
       (GO LP)))
    (replace FTPGENPLIST of GENSTATE with PLIST)
    (SETUPTIMER \FTP.IDLE.TIMEOUT (fetch FTPBUSY of FTPCONNECTION))
    [RETURN (OR NAME (AND FTPDEBUGFLG (HELP "Uninterpretable filename returned by ENUMERATE" PLIST]
  BROKEN
    (ERROR "File server broke connection before directory enumeration finished. RETURN() to terminate
           enumeration." (fetch FTPHOST of FTPCONNECTION))
    (RETURN NIL])

```

(\FTP.FILEINFOFN

```

[LAMBDA (GENSTATE ATTRIBUTE) ; (* bvm%: "26-Apr-84 15:22")
  (\FTP.GETFILEINFO.FROM.PROPS (fetch FTPGENPLIST of GENSTATE)
  ATTRIBUTE])

```

(\FTP.GETFILEINFO

```

[LAMBDA (STREAM ATTRIBUTE DEV) ; Edited 19-Aug-88 11:54 by bvm
  (\FTP.GETFILEINFO.FROM.PROPS [COND
    ((type? STREAM STREAM)
     (fetch FTPFILEPROPS of STREAM))
    (T (\FTP.RECOGNIZEFILE STREAM DEV 'PROPS 'OLD (LIST (\FTP.FROM.LISP.ATTRIBUTE
                                                         ATTRIBUTE]
  ATTRIBUTE])

```

(\FTP.GETFILEINFO.FROM.PROPS

```

[LAMBDA (PROPS ATTRIBUTE) ; (* bvm%: " 5-May-84 16:31")
  (PROG (TMP)

```

```
(RETURN (SELECTQ ATTRIBUTE
(CREATIONDATE (CADR (ASSOC 'CREATION-DATE PROPS)))
(WRITEDATE (CADR (ASSOC 'WRITE-DATE PROPS)))
(READDATE (CADR (ASSOC 'READ-DATE PROPS)))
(ICREATIONDATE
  (IDATE (CADR (ASSOC 'CREATION-DATE PROPS))))
(IWRITEDATE (IDATE (CADR (ASSOC 'WRITE-DATE PROPS))))
(IREADDATE (IDATE (CADR (ASSOC 'READ-DATE PROPS))))
(LENGTH (MKATOM (CADR (ASSOC 'SIZE PROPS))))
(SIZE (AND (SETQ TMP (CADR (ASSOC 'SIZE PROPS)))
  (FIXP (SETQ TMP (MKATOM TMP)))
  (FOLDHI TMP BYTESPERPAGE))))
(TYPE [MKATOM (U-CASE (CADR (ASSOC ATTRIBUTE PROPS))]
(BYTESIZE (MKATOM (CADR (ASSOC 'BYTE-SIZE PROPS))))
(CADR (ASSOC ATTRIBUTE PROPS]))
```

(\FTP.FROM.LISP.ATTRIBUTE

[LAMBDA (ATTR) ; Edited 19-Aug-88 11:48 by bvm

;; Returns FTP name for the specified Lisp attribute, or the attribute itself if unknown.

```
(SELECTQ ATTR
(BYTESIZE 'BYTE-SIZE)
(LENGTH 'SIZE)
((CREATIONDATE ICREATIONDATE)
 'CREATION-DATE)
((WRITEDATE IWRITEDATE)
 'WRITE-DATE)
((READDATE IREADDATE)
 'READ-DATE)
(EOLCONVENTION
 'END-OF-LINE-CONVENTION)
ATTR])
```

)

(RPAQ? \FTPAVAILABLE)

(RPAQ? \FTP.IDLE.TIMEOUT 120000)

(RPAQ? *FTP-IGNORE-SERVER-FULL*)

;; internal

(DEFINEQ

(\FTP.OPEN.CONNECTION

[LAMBDA (HOST EHOSTREAM FAILURESTRING) (* bvm%: " 6-Oct-86 13:57")

```
(LET ((PORT (BESTPUPADDRESS HOST PROMPTWINDOW))
INSTREAM)
```

```
(if [AND PORT (SETQ INSTREAM (OPENBSPSTREAM [CONS (CAR PORT)
(COND
  ((EQ (CDR PORT)
  0)
  \PUPSOCKET.FTP)
  (T (CDR PORT)]
NIL
(FUNCTION \FTP.ERRORHANDLER)
NIL NIL (FUNCTION \FTP.WHENCLOSED)
(OR FAILURESTRING "Can't open FTP connection"]
```

```
then (if (TYPENAMEP INSTREAM 'STREAM)
then (SETQ INSTREAM (create FTPCONNECTION
FTPIN _ INSTREAM
FTPOUT _ (BSPOUTPUTSTREAM INSTREAM)
FTPHOST _ [\CANONICAL.HOSTNAME (COND
  ((LITATOM HOST)
  HOST)
  (T (ETHERHOSTNAME PORT)
  FTPBUSY _ T))
(if (\FTP.SENDVERSION INSTREAM EHOSTREAM)
then (push \FTP.CONNECTIONS INSTREAM)
INSTREAM
else (CLOSEBSPSTREAM (fetch FTPIN of INSTREAM)))
else INSTREAM])
```

(FTP.BREAKCONNECTION

[LAMBDA (HOST IDLEONLY) (* bvm%: "28-Apr-85 14:51")

```
(LET (HOSTS)
(for STREAM in [for CONN in \FTP.CONNECTIONS collect (pushnew HOSTS (fetch FTPHOST of CONN))
(fetch FTPIN of CONN)
```

```
when (AND (OR (EQ HOST T)
(EQ HOST (fetch FTPHOST of CONN)))
(OR (NULL IDLEONLY)
(NULL (fetch FTPBUSY of CONN]
```

```
do (CLOSEBSPSTREAM STREAM 5000))
HOSTS])
```

(FTP.SENDVERSION

; Edited 11-Jan-88 14:54 by bvm

```
[LAMBDA (CONNECTION EHOSTREAM)
  (PROG ((INS (fetch FTPIN of CONNECTION))
        (OUTS (fetch FTPOUT of CONNECTION)))
    (FTPPUTMARK OUTS (MARK# VERSION))
    (BOUT OUTS \FTP.VERSION)
    (PRIN3 "Xerox Lisp Ftp user" OUTS)
    (.EOC. OUTS)
    (RETURN (SELECTC (FTPGETMARK INS)
                    (MARK# VERSION)
                    (COND
                     ((EQ (BIN INS)
                          \FTP.VERSION)
                      (\FTP.FLUSH.TO.EOC INS EHOSTREAM))))
            NIL]))
```

(FTP.WHENCLOSED

(* bvm%: "15-SEP-83 23:06")

```
[LAMBDA (INSTREAM)
  (PROG [(CONN (find c in \FTP.CONNECTIONS suchthat (EQ (fetch FTPIN of C)
                                                         INSTREAM)]
        (COND
         (CONN (SETQ \FTP.CONNECTIONS (DREMOVE CONN \FTP.CONNECTIONS))
                (AND FTPDEBUGFLG (printout FTPDEBUGLOG T "{FTP Connection with " (fetch FTPHOST of CONN)
                " closed}" T))
```

(GETFTP.CONNECTION

; Edited 17-Nov-87 14:55 by bvm:

```
[LAMBDA (HOST UNWINDSAVE TRYHARD)
  (PROG ([H (\CANONICAL.HOSTNAME (if (LITATOM HOST)
                                     then HOST
                                     else (ETHERHOSTNAME HOST])
    CONNECTION)
    (if *FTP-IGNORE-SERVER-FULL*
        then (SETQ TRYHARD T))
  RETRY
    (RETURN (if [SETQ CONNECTION (OR (for CONN in \FTP.CONNECTIONS
                                     when (AND (EQ (fetch FTPHOST of CONN)
                                                  H)
                                              (NOT (fetch FTPBUSY of CONN))
                                              (BSOPENP (fetch FTPIN of CONN)
                                                       'OUTPUT))
                                     do (replace FTPBUSY of CONN with T)
                                     (replace ACCESS of (fetch FTPIN of CONN) with 'INPUT)
                                     ; Because \CLOSEFILE clobbered this field
                                     (replace ACCESS of (fetch FTPOUT of CONN) with 'OUTPUT)
                                     (RETURN CONN))
                    (\FTP.OPEN.CONNECTION HOST (.FTPDEBUGLOG.)
                    (AND TRYHARD 'RETURN]
    then (if (type? FTP.CONNECTION CONNECTION)
            then [if UNWINDSAVE
                  then (RESETSAVE (PROGN (fetch FTPIN of CONNECTION)
                                         ' (AND RESETSTATE (CLOSEBSPSTREAM OLDVALUE 0)
                  CONNECTION
                  (AND TRYHARD (STRINGP CONNECTION))
                  then ; Didn't get connection, but got error return of some sort (normal
                    ; prompt message was suppressed).
                    (if (STRPOS "FULL" CONNECTION NIL NIL NIL NIL UPPERCASEARRAY)
                        then ; Got "server full" abort, try again later
                            (BLOCK 5000)
                            (GO RETRY)
                    else ; Some other problem--print in prompt window before giving up.
                        (CL:FORMAT PROMPTWINDOW "~%%Couldn't get FTP connection to ~A because:
                        ~A" H CONNECTION)
                    NIL]))
```

(RELEASE.FTP.CONNECTION

(* bvm%: "18-MAY-83 10:53")

```
[LAMBDA (CONN)
  (replace FTPBUSY of CONN with NIL)]
```

(FTP.ERRORHANDLER

; Edited 26-Nov-86 15:39 by bvm:

```
[LAMBDA (INSTREAM ERRCODE)
  (PROG (OUTSTREAM TMP)
    (RETURN (SELECTQ ERRCODE
                    (MARK.ENCOUNTERED)
                    (COND
                     ((fetch FTPOPENP of INSTREAM) ; If reading a file, this is EOF
                      (STREAMOP 'ENDOFSTREAMOP INSTREAM INSTREAM))
                     (T -1)))
    (BAD.STATE.FOR.BOUT)
    (COND
     ((AND (SETQ OUTSTREAM (BSOUTPUTSTREAM INSTREAM))
            (fetch FTPOPENP of OUTSTREAM))
```

```

; Writing a file, and partner timed out. Hard to recover from this
(ERROR "File server has broken connection" (fetch FULLFILENAME of OUTSTREAM))
(T ; Just protocol stuff. Let it go by, and catch the error on the next
; input
NIL))
(BAD.STATE.FOR.BIN
(COND
((fetch FTPOPENP of INSTREAM) ; Could recover by reopening file
(\FTP.FIX.BROKEN.INPUT INSTREAM))
((SETQ TMP (STKPOS 'READPLIST)) ; Reading a plist, can't just barf in the middle
(RETFROM TMP NIL T))
(T ; Act like end of file
-1)))
(BAD.GETMARK (COND
((BSPOPENP INSTREAM 'INPUT)
(MARK# NOTAMARK))
(T (MARK# BROKEN))))
(ERROR ERRCODE (AND INSTREAM (OR (fetch FULLFILENAME of INSTREAM)
(AND (SETQ OUTSTREAM (BSPOUTPUTSTREAM INSTREAM))
(fetch FULLFILENAME of OUTSTREAM))
(AND (SETQ OUTSTREAM (BSPFRNADDRESS INSTREAM))
(ETHERHOSTNAME OUTSTREAM T]))))

```

(\FTP.FIX.BROKEN.INPUT

[LAMBDA (INSTREAM) (* bvm%: "28-Apr-85 14:15")

;; Called when remote server breaks connection in midstream. Try to reopen and set fileptr to the right place

```

(PROG ((FULLNAME (fetch FULLFILENAME of INSTREAM))
(PROPS (fetch FTPFILEPROPS of INSTREAM))
(POS (GETFILEPTR INSTREAM))
NEWSTREAM)
(printout PROMPTWINDOW T "File server broke connection while reading " FULLNAME " at byte " .P2 POS
' |...|)
(COND
((SETQ NEWSTREAM (\FTP.OPENFILE.FROM.PLIST (FILENAMEFIELD FULLNAME 'HOST)
[\FTP.ADD.USERINFO (for PAIR in PROPS collect PAIR
when (FMEMB (CAR PAIR)
' (NAME-BODY VERSION DIRECTORY DEVICE
SERVER-FILENAME)
' INPUT))
(\SMASHBSPSTREAM NEWSTREAM INSTREAM) ; Smash new stream into old, so we are now using INSTREAM
; again
(for CONN in \FTP.CONNECTIONS when (EQ (fetch FTPIN of CONN)
NEWSTREAM)
do (replace FTPIN of CONN with INSTREAM)
(replace FTPOUT of CONN with (BSPOUTPUTSTREAM INSTREAM))
(RETURN))
(\BSP.DECLARE.FILEPTR INSTREAM 0)
(printout PROMPTWINDOW T "Reopening file and restoring fileptr...")
(SETFILEPTR INSTREAM POS)
(printout PROMPTWINDOW "done.")
(RETURN T))
(T (ERROR "File server broke connection; unable to reestablish" FULLNAME]))

```

(\FTP.CLEANUP

[LAMBDA NIL (* bvm%: "19-AUG-83 16:19")

;; Process that sits watching to see if an FTP connection has been idle too long

```

(DECLARE (SPECVARS CONNS FAIL))
(PROG ((TIMER (SETUPTIMER 0))
(INTERVAL (LRSH \FTP.IDLE.TIMEOUT 1))
CONNS)
SLEEP
(SETUPTIMER INTERVAL TIMER)
(do (BLOCK NIL TIMER) until (TIMEREXPIRED? TIMER))
LP1 (COND
((NULL (SETQ CONNS \FTP.CONNECTIONS))
(RETURN)))
LP2 (COND
([AND (FIXP (fetch FTPBUSY of (CAR CONNS)))
(TIMEREXPIRED? (fetch FTPBUSY of (CAR CONNS)))
(NOT (PROG (FAIL)
[MAP.PROCESSES (FUNCTION (LAMBDA (PROC)
(COND
((EQ (PROCESS.EVALV PROC 'FTP.CONNECTION)
(CAR CONNS))
(SETQ FAIL T)
(RETURN FAIL]
;; Timer expired AND there is nobody actively using this connection. Latter is important in case the remote server was just slow to
;; answer. Ideal solution would be to see if anyone has a pointer to the generator, but that takes gc changes
(CLOSEBSPSTREAM (fetch FTPIN of (CAR CONNS)))
(GO LP1)))
(COND
((SETQ CONNS (CDR CONNS))

```

(GO LP2)))
(GO SLEEP])

(\FTP.ASSURE.CLEANUP

(* bvm%: "19-AUG-83 16:12")

[LAMBDA NIL
(OR (FIND.PROCESS '\FTP.CLEANUP)
(ADD.PROCESS '\FTP.CLEANUP)
'RESTARTABLE
'NO])

)

(ADDTTOVAR \FTP.CONNECTIONS)

(DEFINEQ

(\FTP.HANDLE.NO

[LAMBDA (CONNECTION BADPLIST ECHOSTREAM CODE LEAVEMARK NOERRORFLG DESTPLIST)

; Edited 17-Nov-87 18:17 by bvm:

(PROG ((INSTREAM (fetch FTPIN of CONNECTION))
(HOST (fetch FTPHOST of CONNECTION))
[FLUSHER (COND

(LEAVEMARK (FUNCTION \FTP.FLUSH.TO.MARK))
(T (FUNCTION \FTP.FLUSH.TO.EOC]

INFO CPASS CNAME NEWNAME)
(SELECTC (OR CODE (SETQ CODE (FTPGETCODE INSTREAM T)))

(\NO.FILE.NOT.FOUND
(APPLY* FLUSHER INSTREAM (OR ECHOSTREAM (.FTPDEBUGLOG.)))
(RETURN))

((LIST \NO.BAD.TRANSFER.PARMS \NO.BAD.EOLCONVENTION)

(COND
((AND (SETQ INFO (ASSOC 'END-OF-LINE-CONVENTION BADPLIST))
(NEQ (CADR INFO)
'CR))

(RPLACA (CDR INFO)
'CR)

; Fall back on EOL = CR, which everyone must support

(APPLY* FLUSHER INSTREAM (OR ECHOSTREAM (.FTPDEBUGLOG.)))
(RETURN T))))

(\NO.FILE.PROTECTED
;; We are very dependent on precedence of errors. The code here assumes that if I specified
;; CONNECT-NAME and still got a protection error, then there is nothing I can do--had the name been
;; illegal or it required a password, I would have gotten a connect name/pass error instead. This is
;; especially a problem for the RENAME command, since the error return gives no indication as to
;; whether it is the source or the destination that has the problem. We solve it here by making the source
;; good first, then if we still get errors, the problem must be in the destination.

(if (COND
(NULL (SETQ CNAME (ASSOC 'CONNECT-NAME BADPLIST))))

; First time thru--try adding connect name for the main file.

[NCONC1 BADPLIST (LIST 'CONNECT-NAME (\FTP.DIRECTORYNAMEONLY
(CADR (ASSOC 'DIRECTORY BADPLIST]

T)

([AND DESTPLIST (NOT (STRING-EQUAL (CADR CNAME)
(SETQ NEWNAME (\FTP.DIRECTORYNAMEONLY
(CADR (ASSOC 'DIRECTORY DESTPLIST]

; Destination directory is different than source--maybe that's the
; problem. Oddly enough, IFS doesn't let us give separate
; connect passwords for the source and destination, so have to
; do it all in main plist.

(RPLACA (CDR CNAME)
NEWNAME)

(if (SETQ CPASS (ASSOC 'CONNECT-PASSWORD BADPLIST))
then (RPLACA (CDR CPASS)
""))

T))

then ; Flush the error message, try again

(APPLY* FLUSHER INSTREAM (OR ECHOSTREAM (.FTPDEBUGLOG.)))
(RETURN T))

(\NO.ILLEGAL.CONNECTPASSWORD ; Connect Password error

[SETQ CNAME (MKATOM (CADR (ASSOC 'CONNECT-NAME BADPLIST]
(if (AND (NULL (SETQ CPASS (ASSOC 'CONNECT-PASSWORD BADPLIST))
(SETQ INFO (\INTERNAL/GETPASSWORD HOST NIL CNAME)))

then ; quietly get a connect password and try again. Thus usually
; gets us the null password unless a real one has been
; previously cached.

(NCONC1 BADPLIST (LIST 'CONNECT-PASSWORD (CDR INFO)))
(APPLY* FLUSHER INSTREAM (OR ECHOSTREAM (.FTPDEBUGLOG.)))
(RETURN T)

else ; Fall thru to process the noisy way

)

NIL)

GENERAL.FAILURE

(printout [OR ECHOSTREAM (SETQ ECHOSTREAM (GETSTREAM PROMPTWINDOW 'OUTPUT]
T HOST " : ")

(COND

((APPLY* FLUSHER INSTREAM ECHOSTREAM)
(SELECTC CODE

```

((LIST \NO.ILLEGAL.USERNAME \NO.ILLEGAL.USERPASSWORD)
; User Password errors
(RETURN (COND
((SETQ INFO (\INTERNAL/GETPASSWORD HOST T NIL NIL))
(for PAIR in BADPLIST do (SELECTQ (CAR PAIR)
(USER-NAME (FRPLACA (CDR PAIR)
(CAR INFO)))
(USER-PASSWORD
(FRPLACA (CDR PAIR)
(CDR INFO)))
NIL) )
T))))
(\NO.ILLEGAL.CONNECTPASSWORD ; Connect Password error
[RETURN
(COND
((SETQ INFO (\INTERNAL/GETPASSWORD HOST T CNAME NIL))
[COND
(CPASS (FRPLACA (CDR CPASS)
(CDR INFO)))
(T (NCONC1 BADPLIST (LIST 'CONNECT-PASSWORD (CDR INFO)
T)
(AND (NOT NOERRORFLG)
(LISPERROR "PROTECTION VIOLATION"
(\FTP.PACKFILENAME
HOST
(if [OR (NULL DESTPLIST)
(String-EQUAL CNAME (\FTP.DIRECTORYNAMEONLY
(CADR (ASSOC 'DIRECTORY BADPLIST]
then BADPLIST
else ; Problem is probably with the destination
DESTPLIST)
NIL T])
(\NO.ILLEGAL.NAME.ERRORS
(OR NOERRORFLG (LISPERROR "BAD FILE NAME" (\FTP.PACKFILENAME HOST BADPLIST NIL T))))
(\NO.STORAGE.FULL
(OR NOERRORFLG (LISPERROR "FILE SYSTEM RESOURCES EXCEEDED" (\FTP.PACKFILENAME HOST
BADPLIST NIL T))))
(\NO.FILE.PROTECTED
(OR NOERRORFLG (LISPERROR "PROTECTION VIOLATION" (\FTP.PACKFILENAME HOST BADPLIST NIL T)
))
(GO WONT.OPEN)))
(T (\FTPERROR CONNECTION)))
(RETURN)
WONT.OPEN
(OR NOERRORFLG (LISPERROR "FILE WON'T OPEN" (\FTP.PACKFILENAME HOST BADPLIST NIL T])

```

```

(\FTP.DIRECTORYNAMEONLY ; Edited 16-Nov-87 16:09 by bvm
[LAMBDA (DIRNAME)
(LET ((N (STREPOS '> DIRNAME)))
(COND
(N (SUBSTRING DIRNAME 1 (- N 1)))
(T DIRNAME]))

```

```

(\FTP.EOL.FROM.PLIST (* bvm%: "21-NOV-83 15:33")
[LAMBDA (PLIST)
(for PAIR in PLIST when (EQ (CAR PAIR)
'END-OF-LINE-CONVENTION)
do (RETURN (SELECTQ (CADR PAIR)
(LF LF.EOLC)
(CRLF CRLF.EOLC)
CR.EOLC))
finally (RETURN CR.EOLC])

```

```

(\FTP.MAKEPLIST (* bvm%: "4-JUN-83 21:35")
[LAMBDA (FILENAME HOST DESIREDPROPS)
(PROG ((INFO (\INTERNAL/GETPASSWORD HOST))
(RETURN (CONS (LIST 'USER-NAME (CAR INFO))
(CONS (LIST 'USER-PASSWORD (CDR INFO))
(CONS (LIST 'SERVER-FILENAME FILENAME)
(for PROP inside DESIREDPROPS collect (LIST 'DESIRED-PROPERTY PROP))

```

```

(\FTP.PRINTPLIST ; Edited 11-Jan-88 16:09 by bvm
[LAMBDA (STREAM PLIST)
(BOUT STREAM (CHARCODE %))
(for PAIR in PLIST do (for ITEM in PAIR bind (BEFORE _ (CHARCODE %))
ISPASSWORD
do (BOUT STREAM BEFORE)
(SETQ BEFORE (CHARCODE SPACE))
[for CH inpname ITEM do (SELCHARQ (COND
(ISPASSWORD (SETQ CH (\DECRYPT.PWD.CHAR CH))
(T CH))
((% ( %))
(BOUT STREAM (CHARCODE %'))))

```


(\FTP.ADD.QUOTES

[LAMBDA (NAME) ; Edited 11-Jan-88 16:52 by bvm

;; The only funny char we know about is quote, so quote all the quotes with a quote.

```
(bind (N _ 1)
  I PIECES while (SETQ I (STRPOS "'" NAME N)) do (push PIECES "'" (SUBSTRING NAME N I))
  (SETQ N (ADD1 I))
  finally (RETURN (if (PIECES
    then (if (<= N (NCHARS NAME))
      then (push PIECES (SUBSTRING NAME N))
      (CONCATLIST (DREVERSE PIECES))
    else
      NAME])) ; nothing got quoted
```

(\FTP.PACK.DIRECTORYNAMEP

[LAMBDA (CONNECTION PLIST) (* lmm "25-Mar-85 14:38")

```
(PROG [(DIRECTORY (CADR (ASSOC 'DIRECTORY PLIST)))
  (DEVICE (CADR (ASSOC 'DEVICE PLIST))
  (RETURN (PACKFILENAME.STRING 'HOST (fetch FTPHOST of CONNECTION)
    'DEVICE DEVICE 'DIRECTORY DIRECTORY]))
```

(\FTP.UNPACKFILENAME

[LAMBDA (FILENAME NOLOGIN) ; Edited 16-Nov-87 17:43 by bvm

```
(LET ((FIELDS (UNPACKFILENAME.STRING FILENAME))
  PLIST HOST DEVICE DIR NAME EXT INFO)
  (for TAIL on FIELDS by (CDDR TAIL) do (SELECTQ (CAR TAIL)
    (HOST (SETQ HOST (CADR TAIL)))
    (DIRECTORY (SETQ DIR (CADR TAIL)))
    (DEVICE (SETQ DEVICE (CADR TAIL)))
    (NAME (SETQ NAME (CADR TAIL)))
    (EXTENSION (SETQ EXT (CADR TAIL)))
    (VERSION (push PLIST (LIST 'VERSION (CADR TAIL))))
    (NIL)))
```

```
(COND
  ((AND HOST (SETQ HOST (\CANONICAL.HOSTNAME HOST)))
  [push PLIST (LIST 'NAME-BODY (COND
    ((AND EXT (> (NCHARS EXT)
      0))
    (CONCAT NAME '%. EXT))
    (T NAME])
```

```
[COND
  (DIR [COND
    ((EQ (GETHOSTINFO HOST 'OSTYPE)
      'UNIX)
    [COND
      ((EQ (NTHCHARCODE DIR -1)
        (CHARCODE /))
      (SETQ DIR (SUBSTRING DIR 1 -2])
    [COND
      ((NEQ (NTHCHARCODE DIR 1)
        (CHARCODE /))
      (SETQ DIR (CONCAT '/' DIR])
    (COND
      ((U-CASEP DIR)
      (SETQ DIR (L-CASE DIR)
      (push PLIST (LIST 'DIRECTORY DIR])
    [COND
      (DEVICE (push PLIST (LIST 'DEVICE DEVICE])
      (CONS HOST (if NOLOGIN
        then PLIST
        else (\FTP.ADD.USERINFO PLIST HOST]))
```

; Coerce directory name to lowercase, get rid of trailing /

(\FTP.ADD.USERINFO

[LAMBDA (PLIST HOST) (* bvm%: "27-OCT-83 15:50")

```
(PROG ((INFO (\INTERNAL/GETPASSWORD HOST)))
  (push PLIST (LIST 'USER-NAME (CAR INFO))
  (LIST 'USER-PASSWORD (CDR INFO)))
  (RETURN PLIST))
```

(\FTP.FLUSH.TO.EOC

[LAMBDA (INSTREAM EHOSTREAM) (* bvm%: "13-JUN-83 15:36")

;; Eat bytes from the input side of CONNECTION up to next mark, copying bytes to EHOSTREAM if given, and return T if the mark is EOC

```
(PROG ([STREAM (AND EHOSTREAM (GETSTREAM EHOSTREAM 'OUTPUT]
  CH)
  (while (NEQ (SETQ CH (BIN INSTREAM))
    -1)
  do (AND STREAM (\OUTCHAR STREAM CH)))
  (RETURN (EQ (FTPGETMARK INSTREAM)
    (MARK# EOC]))
```

(\FTP.FLUSH.TO.MARK


```
[LAMBDA (INSTREAM EHOSTREAM) (* bvm%: "7-JUL-83 12:08")
  (bind CH [STREAM _ (AND EHOSTREAM (GETSTREAM EHOSTREAM 'OUTPUT] while (NEQ (SETQ CH (BIN INSTREAM))
    -1)
  do (AND STREAM (\OUTCHAR STREAM CH)))
T])
```

(\FTPERROR

```
[LAMBDA (CONNECTION ERRMSG ERRARG) (* bvm%: "11-Jul-84 15:33")
  (COND
    (FTPDEBUGFLG (printout FTPDEBUGLOG T "{FTP Protocol violation, aborted}" T)
      (HELP)))
  (CLOSEBSPSTREAM (COND
    ((type? STREAM CONNECTION)
      CONNECTION)
    (T (fetch FTPIN of CONNECTION))))
  1000)
(COND
  (ERRMSG (ERROR (COND
    ((EQ ERRMSG T)
      "FTP Protocol violation")
    (T ERRMSG))
    ERRARG])
)
```

)

:: for debugging

(DEFINEQ

(FTPDEBUG

```
[LAMBDA (FLG REGION) ; Edited 16-Nov-87 16:13 by bvm
  (SETQ FTPDEBUGLOG (CREATEW REGION "FTP Debug info"))
  [WINDOWPROP FTPDEBUGLOG 'CLOSEFN (FUNCTION (LAMBDA (WINDOW)
    (AND (EQ (WINDOWPROP WINDOW 'DSP)
      FTPDEBUGLOG)
      (SETQ FTPDEBUGLOG (SETQ FTPDEBUGFLG NIL])
  [WINDOWPROP FTPDEBUGLOG 'SHRINKFN (FUNCTION (LAMBDA (WINDOW)
    ;; Suspend tracing while window shrunk
    (AND (EQ (WINDOWPROP WINDOW 'DSP)
      FTPDEBUGLOG)
      (SETQ FTPDEBUGFLG NIL])
  [WINDOWPROP FTPDEBUGLOG 'EXPANDFN (FUNCTION (LAMBDA (WINDOW)
    ;; Turn back on when expanded
    (AND (EQ (WINDOWPROP WINDOW 'DSP)
      FTPDEBUGLOG)
      (SETQ FTPDEBUGFLG T])
  (SETQ FTPDEBUGLOG (WINDOWPROP FTPDEBUGLOG 'DSP))
  (DSPFONT (FONTCREATE 'GACHA 8)
    FTPDEBUGLOG)
  (DSPSCROLL T FTPDEBUGLOG)
  (SETQ FTPDEBUGFLG T])
```

(FTPPRINTMARK

```
[LAMBDA (MARK) (* bvm%: "25-Aug-84 21:58")
  [COND
    (FTPDEBUGFLG (printout FTPDEBUGLOG "[" (OR (CADR (FASSOC MARK (LISTP FTPMARKTYPES)))
      MARK)
      "]"")
    (COND
      ((EQ MARK (MARK# EOC))
        (TERPRI FTPDEBUGLOG]
  MARK])
```

(FTPPRINTCODE

```
[LAMBDA (CODE NOCODEP) (* bvm%: "20-AUG-83 00:12")
  (COND
    (FTPDEBUGFLG (PRIN1 '{ FTPDEBUGLOG)
      (COND
        (NOCODEP (PRINTCONSTANT CODE FTPNOCODES FTPDEBUGLOG "\NO."))
        (T (PRIN1 '{ (FIX 1)
          CODE FTPDEBUGLOG)))
      (PRIN1 '{ } FTPDEBUGLOG)))
  CODE])
```

(FTPGETMARK

```
[LAMBDA (STREAM) (* bvm%: "23-Nov-86 15:50")
  (bind MARK while (EQ (SETQ MARK (FTPPRINTMARK (BSPGETMARK STREAM)))
    (MARK# COMMENT))
  do (\FTP.FLUSH.TO.MARK STREAM (.FTPDEBUGLOG.)) finally (RETURN MARK])
```

(FTPPUTMARK

[LAMBDA (STREAM MARK)
(BSPPUTMARK STREAM (**FTPPRINTMARK** MARK))

(* bvm%: "12-MAY-83 10:24")

(FTPPUTCODE

[LAMBDA (STREAM CODE NOCODEP)
(BOUT STREAM (**FTPPRINTCODE** CODE NOCODEP))

(* bvm%: "20-AUG-83 00:12")

(FTPGETCODE

[LAMBDA (STREAM NOCODEP)
(**FTPPRINTCODE** (BIN STREAM)
NOCODEP)]

(* bvm%: "20-AUG-83 00:17")

)

(RPAQ? **FTPDEBUGLOG**)

(RPAQ? **FTPDEBUGFLG**)

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ **FTPMARKTYPES**

- (1 RETRIEVE)
- (2 STORE)
- (3 YES)
- (4 NO)
- (5 HERE-IS-FILE)
- (6 EOC)
- (7 COMMENT)
- (8 VERSION)
- (9 NEW-STORE)
- (10 ENUMERATE)
- (11 HERE-IS-PLIST)
- (12 NEW-ENUMERATE)
- (14 DELETE)
- (15 RENAME)
- (16 STORE-MAIL)
- (17 RETRIEVE-MAIL)
- (18 FLUSH-MAILBOX)
- (19 MAILBOX-EXCEPTION)
- (253 NOTAMARK)
- (254 BROKEN))

(DECLARE%: EVAL@COMPILE

(RPAQQ **\FTP.VERSION** 1)

(CONSTANTS **\FTP.VERSION**)

)

(RPAQQ **FTPNOCODES**

- ((\NO.UNIMPLEMENTED 1)
- (\NO.PROTOCOL.ERROR 3)
- (\NO.BAD.PLIST 8)
- (\NO.ILLEGAL.DIRECTORY 10)
- (\NO.ILLEGAL.NAME.ERRORS '(9 10 11 12 25))
- (\NO.BAD.EOLCONVENTION 15)
- (\NO.ILLEGAL.USERNAME 16)
- (\NO.ILLEGAL.USERPASSWORD 17)
- (\NO.ILLEGAL.CONNECTNAME 19)
- (\NO.ILLEGAL.CONNECTPASSWORD 20)
- (\NO.FILE.NOT.FOUND 64)
- (\NO.FILE.PROTECTED 65)
- (\NO.BAD.TRANSFER.PARMS 66)
- (\NO.DISK.ERROR 67)
- (\NO.STORAGE.FULL 68)
- (\NO.UNSPECIFIED.ERRORS '(71 72))
- (\NO.FILE.BUSY 73)
- (\NO.RENAME.DESTINATION.EXISTS 74))

(DECLARE%: EVAL@COMPILE

(RPAQQ **\NO.UNIMPLEMENTED** 1)

(RPAQQ **\NO.PROTOCOL.ERROR** 3)

(RPAQQ **\NO.BAD.PLIST** 8)

(RPAQQ **\NO.ILLEGAL.DIRECTORY** 10)

(RPAQQ **\NO.ILLEGAL.NAME.ERRORS** (9 10 11 12 25))

(RPAQQ **\NO.BAD.EOLCONVENTION** 15)

(RPAQQ **\NO.ILLEGAL.USERNAME** 16)

(RPAQQ \NO.ILLEGAL.USERPASSWORD 17)

(RPAQQ \NO.ILLEGAL.CONNECTNAME 19)

(RPAQQ \NO.ILLEGAL.CONNECTPASSWORD 20)

(RPAQQ \NO.FILE.NOT.FOUND 64)

(RPAQQ \NO.FILE.PROTECTED 65)

(RPAQQ \NO.BAD.TRANSFER.PARMS 66)

(RPAQQ \NO.DISK.ERROR 67)

(RPAQQ \NO.STORAGE.FULL 68)

(RPAQQ \NO.UNSPECIFIED.ERRORS (71 72))

(RPAQQ \NO.FILE.BUSY 73)

(RPAQQ \NO.RENAME.DESTINATION.EXISTS 74)

(CONSTANTS (\NO.UNIMPLEMENTED 1)
(\NO.PROTOCOL.ERROR 3)
(\NO.BAD.PLIST 8)
(\NO.ILLEGAL.DIRECTORY 10)
(\NO.ILLEGAL.NAME.ERRORS '(9 10 11 12 25))
(\NO.BAD.EOLCONVENTION 15)
(\NO.ILLEGAL.USERNAME 16)
(\NO.ILLEGAL.USERPASSWORD 17)
(\NO.ILLEGAL.CONNECTNAME 19)
(\NO.ILLEGAL.CONNECTPASSWORD 20)
(\NO.FILE.NOT.FOUND 64)
(\NO.FILE.PROTECTED 65)
(\NO.BAD.TRANSFER.PARMS 66)
(\NO.DISK.ERROR 67)
(\NO.STORAGE.FULL 68)
(\NO.UNSPECIFIED.ERRORS '(71 72))
(\NO.FILE.BUSY 73)
(\NO.RENAME.DESTINATION.EXISTS 74))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS MARK# MACRO [X (OR [CAR (find M in FTPMARKTYPES suchthat (EQ (CADR M) (CAR X])
(HELP "Unknown mark type" (CAR X))

(PUTPROPS .EOC. MACRO ((STREAM)
(FTPPUTMARK STREAM (MARK# EOC)))

(PUTPROPS .FTPDEBUGLOG. MACRO (NIL (AND FTPDEBUGFLG FTPDEBUGLOG)))
)

(PUTPROPS MARK# INFO NOEVAL)

(DECLARE%: EVAL@COMPILE

(RECORD FTPCONNECTION (FTPIN FTPOUT FTPHOST FTPBUSY FTPCURRENTFILE)
(TYPE? LISTP))

(ACCESSFNS FTPSTREAM ((FTPFILEPROPS (fetch F5 of DATUM)
(replace F5 of DATUM with NEWVALUE)))
(SYNONYM FTPFILEPROPS (FTPOPENP)))

(RECORD FTPFILEEGENSTATE (FTPGENCONNECTION FTPGENPLIST FTPDEVICEWANTED FTPNAMEFILTER))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS FTPDEBUGFLG \FTPCONNECTIONS \FTPAVAILABLE \FTP.IDLE.TIMEOUT \BSPFDEV \FTPDEV)
)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\FTPINIT)
)

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

FUNCTION INDEX

FTP.BREAKCONNECTION	10	\FTP.ERRORHANDLER	11	\FTP.PACKFILENAME	15
FTPDEBUG	17	\FTP.FILEINFOFN	9	\FTP.PRINTPLIST	14
FTPGETCODE	18	\FTP.FIX.BROKEN.INPUT	12	\FTP.RECOGNIZEFILE	3
FTPGETMARK	17	\FTP.FLUSH.TO.EOC	16	\FTP.REGISTER	5
FTPPRINTCODE	17	\FTP.FLUSH.TO.MARK	16	\FTP.RENAMEFILE	5
FTPPRINTMARK	17	\FTP.FROM.LISP.ATTRIBUTE	10	\FTP.REOPENFILE	2
FTPPUTCODE	18	\FTP.GENERATEFILES	7	\FTP.SENDVERSION	11
FTPPUTMARK	18	\FTP.GETFILEINFO	9	\FTP.UNPACKFILENAME	16
\FTP.ADD.QUOTES	16	\FTP.GETFILEINFO.FROM.PROPS	9	\FTP.UNREGISTER	5
\FTP.ADD.USERINFO	16	\FTP.GETFILENAME	3	\FTP.WHENCLOSED	11
\FTP.ASSURE.CLEANUP	13	\FTP.HANDLE.NO	13	\FTPERROR	17
\FTP.CLEANUP	12	\FTP.MAKEPLIST	14	\FTP.EVENTFN	1
\FTP.CLOSEFILE	4	\FTP.NEXTFILE	9	\FTPINIT	1
\FTP.DELETEFILE	6	\FTP.OPEN.CONNECTION	10	\GETFTPCONNECTION	11
\FTP.DIRECTORYNAMEONLY	14	\FTP.OPENFILE	2	\RELEASE.FTPCONNECTION	11
\FTP.DIRECTORYNAMEP	4	\FTP.OPENFILE.FROM.PLIST	2		
\FTP.EOL.FROM.PLIST	14	\FTP.PACK.DIRECTORYNAMEP	16		

CONSTANT INDEX

\FTP.VERSION	18	\NO.FILE.PROTECTED	19	\NO.PROTOCOL.ERROR	19
\NO.BAD.EOLCONVENTION	19	\NO.ILLEGAL.CONNECTNAME	19	\NO.RENAME.DESTINATION.EXISTS	19
\NO.BAD.PLIST	19	\NO.ILLEGAL.CONNECTPASSWORD	19	\NO.STORAGE.FULL	19
\NO.BAD.TRANSFER.PARMS	19	\NO.ILLEGAL.DIRECTORY	19	\NO.UNIMPLEMENTED	19
\NO.DISK.ERROR	19	\NO.ILLEGAL.NAME.ERRORS	19	\NO.UNSPECIFIED.ERRORS	19
\NO.FILE.BUSY	19	\NO.ILLEGAL.USERNAME	19		
\NO.FILE.NOT.FOUND	19	\NO.ILLEGAL.USERPASSWORD	19		

VARIABLE INDEX

FTP-IGNORE-SERVER-FULL	10	FTPMARKTYPES	18	\FTPAVAILABLE	10
FTPDEBUGFLG	18	FTPNOCODES	18	\FTPCONNECTIONS	13
FTPDEBUGLOG	18	\FTP.IDLE.TIMEOUT	10		

RECORD INDEX

FTPCONNECTION	19	FTPFILEGENSTATE	19	FTPSTREAM	19
---------------------	----	-----------------------	----	-----------------	----

MACRO INDEX

.EOC.	19	.FTPDEBUGLOG.	19	MARK#	19
------------	----	--------------------	----	-------------	----

PROPERTY INDEX

MARK#	19
-------------	----
