

File created: 13-Sep-90 15:07:59 {DSK}<LISPFILERS>TCP>TCPFTPSRV.;6

changes to: (FNS TCPFTP.SERVER.MERGE.PATHNAMES TCPFTP.SERVER.PATH TCPFTP.SERVER.LIST TCPFTP.SERVER.RETRIEVE)

previous date: 11-Sep-90 13:34:33 {DSK}<LISPFILERS>TCP>TCPFTPSRV.;5

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

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

(RPAQQ **TCPFTPSRVCOMS**

```
((FNS TCPFTP.SERVER TCPFTP.SERVER.ABORTED TCPFTP.SERVER.ACCOUNT TCPFTP.SERVER.APPEND
TCPFTP.SERVER.CLOSE.DATA.CONNECTION TCPFTP.SERVER.COMMAND.LOOP TCPFTP.SERVER.CONNECTED.INFO
TCPFTP.SERVER.DELETE TCPFTP.SERVER.DIRECTORY TCPFTP.SERVER.EXIT TCPFTP.SERVER.IDLE.INFO
TCPFTP.SERVER.LIST TCPFTP.SERVER.MERGE.PATHNAMES TCPFTP.SERVER.MODE
TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTP.SERVER.PARSE.PORT TCPFTP.SERVER.PASSWORD
TCPFTP.SERVER.PATH TCPFTP.SERVER.PORT TCPFTP.SERVER.PROCESS TCPFTP.SERVER.RENAME.FROM
TCPFTP.SERVER.RENAME.TO TCPFTP.SERVER.RESPONSE TCPFTP.SERVER.RETRIEVE TCPFTP.SERVER.RETURN.FILE
TCPFTP.SERVER.STORE TCPFTP.SERVER.STRUCTURE TCPFTP.SERVER.TYPE TCPFTP.SERVER.USER
TCPFTP.SERVER.VERBOSE.LIST TCPFTP.SERVER.WAIT.FOR.IDLE TCPFTP.UNIX.LS.DATE)
(INITVARS (TCPFTP.SERVER.HERALD.STRING "Venue Medley FTP Service 1.0 at your service")
(TCPFTP.SERVER.USE.TOPS20.SYNTAX NIL)
(TCPFTP.SERVER.RETRYCOUNT 5))
(GLOBALVARS TCPFTP.SERVER.HERALD.STRING TCPFTP.SERVER.USE.TOPS20.SYNTAX TCPFTP.SERVER.RETRYCOUNT)
(FILE (SYSLOAD)
TCPFTP)))
```

(DEFINEQ

(**TCPFTP.SERVER**

[LAMBDA (PORT DEFAULT.FILE.PATH)

; Edited 24-Aug-87 17:57 by scp

(* * This is the TCP-based FTP server top-level)

```
(ADD.PROCESS `(TCPFTP.SERVER.PROCESS ,PORT ,DEFAULT.FILE.PATH)
'RESTARTABLE
'HARDRESET])
```

(**TCPFTP.SERVER.ABORTED**

[LAMBDA (INSTREAM OUTSTREAM)
(TCPFTP.SERVER.EXIT INSTREAM OUTSTREAM)]

(* ejs%: "20-Mar-86 19:53")

(**TCPFTP.SERVER.ACCOUNT**

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 11:40")

(* * This function parses USER commands)

```
(LET ((ACCT (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
RDTBL)))
(COND
(FTPDEBUGFLG (printout FTPDEBUGLOG ACCT T)))
(DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
(TCPFTP.SERVER.RESPONSE 230 "You sure are formal!" (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON)))
```

(**TCPFTP.SERVER.APPEND**

[LAMBDA (TCPFTP.CON RDTBL USERPORT TYPE DEFAULT.PATH)

(* ejs%: "24-Mar-86 14:07")

(* * This function parses USER commands)

```
(LET* [(FILENAME (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
RDTBL))
(PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))
(TRUENAME (CAR (NLSETQ (OR (INFILE PACKED.FILENAME)
(OUTFILE PACKED.FILENAME))
(COND
(FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
(DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
(COND
(TRUENAME (LET [(FILESTREAM (CAR (NLSETQ (OPENSTREAM TRUENAME 'APPEND NIL `((TYPE %, TYPE)
(COND
[FILESTREAM (TCPFTP.SERVER.RESPONSE 150 (CONCAT "Opening data connection for
append to " (FULLNAME FILESTREAM
))
(fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
(LET ((DATASTREAM (TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTP.CON USERPORT
T)))
(COND
(DATASTREAM [replace (STREAM ENDOFSTREAMOP) of DATASTREAM
```

```

with (FUNCTION (LAMBDA (STREAM)
  (ERROR!])
  (RESETLIST
    (RESETSAVE (COND
      ((EQ TYPE 'BINARY)
        (COPYBYTES DATASTREAM FILESTREAM))
      (T (COPYCHARS DATASTREAM FILESTREAM)))
    (LIST [FUNCTION (LAMBDA (FILESTREAM TCPFTP.CON)
      (CLOSEF? FILESTREAM)
      TCPFTP.SERVER.CLOSE.DATA.CONNECTION
      TCPFTP.CON)
      (TCPFTP.SERVER.RESPONSE
        226 "Data transfer complete"
        (fetch (TCPFTP.CON TCPOUT)
          of TCPFTP.CON]
      FILESTREAM TCPFTP.CON))])
  (T (TCPFTP.SERVER.RESPONSE 450 (CONCAT "Couldn't open file " TRUENAME)
    (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON]
  (T (TCPFTP.SERVER.RESPONSE 550 (CONCAT "Unable to create output filename - " PACKED.FILENAME)
    (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON])

```

(TCPFTP.SERVER.CLOSE.DATA.CONNECTION

```

[LAMBDA (TCPFTP.CON) (* ejs%: "20-Mar-86 17:53")
  (LET ((DATASTREAM (fetch (TCPFTP.CON DATASTREAM) of TCPFTP.CON))
    (EVENT (fetch (TCPFTP.CON BUSY?) of TCPFTP.CON)))
    (CLOSEF? DATASTREAM)
    (replace (TCPFTP.CON DATASTREAM) of TCPFTP.CON with NIL)
    (replace (TCPFTP.CON BUSY?) of TCPFTP.CON with NIL)
    (NOTIFY.EVENT EVENT)
  T])

```

(TCPFTP.SERVER.COMMAND.LOOP

```

[LAMBDA (CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM PATH) ; Edited 31-Aug-90 17:15 by gadener
  (DECLARE (SPECVARS TCPFTP.CON COMMAND))
  (LET ([COMMAND.RDTBL (DEFERREDCONSTANT (PROG [(R (COPYREADTABLE 'ORIG)
    (SETBRK NIL NIL R)
    (SETSYNTAX '%% 'OTHER R)
    (SETSEPR ' (13 10 31 32)
      NIL R)
    (RETURN R]
    (TCPFTP.CON (create TCPFTP.CON
      TCPIN _ CONTROL.INPUT.STREAM
      TCPOUT _ CONTROL.OUTPUT.STREAM))
    (TYPE TCP.DEFAULTFILETYPE)
    RENAME.FROM.FILE LAST.COMMAND.USERPORT)
    (OR PATH (SETQ PATH "{DSK}<LISPFILES>"))
    (while (AND (OPENP CONTROL.INPUT.STREAM 'INPUT)
      (OPENP CONTROL.OUTPUT.STREAM 'OUTPUT)
      (NOT (EOF? CONTROL.INPUT.STREAM)))
      first [PROCESSPROP (THIS.PROCESS)
        'NAME
        (CONCAT "FTP#" (\IP.ADDRESS.TO.STRING (fetch (TCP.CONTROL.BLOCK TCB.DST.HOST)
          of (fetch (TCPSTREAM TCB) of CONTROL.INPUT.STREAM))
        do (LET [(COMMAND (U-CASE (CAR (NLSETQ (READ CONTROL.INPUT.STREAM COMMAND.RDTBL)
          [COND
            ((AND (OPENP CONTROL.INPUT.STREAM 'INPUT)
              (NOT (EOF? CONTROL.INPUT.STREAM)))
              (COND
                ([NOT (FMEMB COMMAND ' (QUIT REIN ABOR NOOP NIL]
                  (BIN CONTROL.INPUT.STREAM] (* Advance past the space preceding the argument)
              (COND
                (FTPDEBUGFLG (printout FTPDEBUGLOG T "> " COMMAND " ")))
                (SELECTQ COMMAND
                  (USER (TCPFTP.SERVER.USER TCPFTP.CON COMMAND.RDTBL))
                  (PASS (TCPFTP.SERVER.PASSWORD TCPFTP.CON COMMAND.RDTBL))
                  (ACCT (TCPFTP.SERVER.ACCOUNT TCPFTP.CON COMMAND.RDTBL))
                  (CWD (SETQ PATH (OR (TCPFTP.SERVER.PATH TCPFTP.CON COMMAND.RDTBL PATH)
                    PATH)))
                  (PWD (TCPFTP.SERVER.RESPONSE 250 (CONCAT "Default pathname is " PATH)
                    (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON)))
                  (PORT (SETQ USERPORT (OR (TCPFTP.SERVER.PORT TCPFTP.CON COMMAND.RDTBL)
                    USERPORT)))
                  (TYPE (SETQ TYPE (OR (TCPFTP.SERVER.TYPE TCPFTP.CON COMMAND.RDTBL)
                    TYPE)))
                  (MODE (TCPFTP.SERVER.MODE TCPFTP.CON COMMAND.RDTBL))
                  (STRU (TCPFTP.SERVER.STRUCTURE TCPFTP.CON COMMAND.RDTBL))
                  (* ;; "Depending on the COMMAND (LIST -> verbose), TCPFTP.SERVER.LIST will return a
                    verbose listing or a simple list of file names ")
                  ((NLST LIST) ; Depending on the COMMAND, TCPFTP.SERVER.LIST will
                    ; return
                    (TCPFTP.SERVER.LIST TCPFTP.CON COMMAND.RDTBL USERPORT PATH COMMAND))
                  (RETR (TCPFTP.SERVER.RETRIEVE TCPFTP.CON COMMAND.RDTBL USERPORT TYPE PATH))
                  (STOR (TCPFTP.SERVER.STORE TCPFTP.CON COMMAND.RDTBL USERPORT TYPE PATH))
                  (APPE (TCPFTP.SERVER.APPEND TCPFTP.CON COMMAND.RDTBL USERPORT TYPE PATH))

```

```

(DELETE (TCPFTP.SERVER.DELETE TCPFTPCON COMMAND.RDTBL PATH))
(RNFR (SETQ RENAME.FROM.FILE (TCPFTP.SERVER.RENAME.FROM TCPFTPCON COMMAND.RDTBL PATH))
)
(RNTO (COND
      ((EQ LAST.COMMAND 'RNFR)
       (TCPFTP.SERVER.RENAME.TO TCPFTPCON COMMAND.RDTBL PATH RENAME.FROM.FILE))
      (T (TCPFTP.SERVER.RESPONSE 503 "I need a RNFR command immediately preceding a
          RNTO command." CONTROL.OUTPUT.STREAM))))
(REIN (DISCARDLINE CONTROL.INPUT.STREAM)
      (TCPFTP.SERVER.WAIT.FOR.IDLE TCPFTPCON)
      (TCPFTP.SERVER.RESPONSE 220 "Go ahead" CONTROL.OUTPUT.STREAM))
(QUIT (DISCARDLINE CONTROL.INPUT.STREAM)
      (TCPFTP.SERVER.WAIT.FOR.IDLE TCPFTPCON)
      (TCPFTP.SERVER.RESPONSE 221 "It's been real" CONTROL.OUTPUT.STREAM)
      (RETURN))
(NOOB (TCPFTP.SERVER.RESPONSE 200 "I'm still here" CONTROL.OUTPUT.STREAM))
(NIL (* Error reading from control stream))
(ERROR!))
(PROGN (DISCARDLINE CONTROL.INPUT.STREAM)
      (TCPFTP.SERVER.RESPONSE 502 (CONCAT "Unrecognized command " COMMAND)
        CONTROL.OUTPUT.STREAM))
      (SETQ LAST.COMMAND COMMAND))

```

(TCPFTP.SERVER.CONNECTED.INFO

```

[LAMBDA (PROCESS BUTTON)
  [PROMPTPRINT "TCPFTP server connected to " (IPHOSTNAME (fetch (TCP.CONTROL.BLOCK TCB.DST.HOST)
    of (fetch (TCPSTREAM TCB) of CONTROL.INPUT.STREAM])
    (* ejs%: "21-Mar-86 17:07")
  (COND
    ((EQ BUTTON 'MIDDLE)
     (COND
       ((AND (BOUNDP 'TCPFTPCON)
              (fetch (TCPFTPCON BUSY?) of TCPFTPCON))
        (printout PROMPTWINDOW T " Server is busy; last command was " (OR (AND (BOUNDP 'COMMAND)
          COMMAND)
          "???"))))
       ((AND (BOUNDP COMMAND)
              COMMAND)
        (printout PROMPTWINDOW T " Last command was " COMMAND]))
    ))

```

(TCPFTP.SERVER.DELETE

```

[LAMBDA (TCPFTPCON RDTBL DEFAULT.PATH)
  (* ejs%: "7-Apr-86 11:42")
  (* * This function parses USER commands)
  (LET* ((FILENAME (RSTRING (fetch (TCPFTPCON TCPIN) of TCPFTPCON)
    RDTBL))
    (PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))
    TRUENAME)
    (COND
      (FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
    (DISCARDLINE (fetch (TCPFTPCON TCPIN) of TCPFTPCON))
    (COND
      [PACKED.FILENAME (COND
        ([SETQ TRUENAME (CAR (NLSETQ (DELFILE PACKED.FILENAME)
          (TCPFTP.SERVER.RESPONSE 250 (CONCAT "Deleted "
            (COND
              (TCPFTP.SERVER.USE.TOPS20.SYNTAX
                (REPACKFILENAME.STRING
                  (PACKFILENAME.STRING 'HOST NIL
                    'BODY TRUENAME)
                  'TOPS-20))
              (T TRUENAME))))
          (fetch (TCPFTPCON TCPOUT) of TCPFTPCON)))]
        (T (TCPFTP.SERVER.RESPONSE 450 (CONCAT "Couldn't delete file " TRUENAME)
          (fetch (TCPFTPCON TCPOUT) of TCPFTPCON)]
        (T (TCPFTP.SERVER.RESPONSE 550 (CONCAT "File not found - " PACKED.FILENAME)
          (fetch (TCPFTPCON TCPOUT) of TCPFTPCON)]

```

(TCPFTP.SERVER.DIRECTORY

```

[LAMBDA (TCPFTPCON RDTBL USERPORT DEFAULT.PATH COMMAND) ; Edited 30-Aug-90 17:47 by gadener

```

```

(* * This function parses USER commands)

```

```

(LET* [(PATH (RSTRING (fetch (TCPFTPCON TCPIN) of TCPFTPCON)
  RDTBL))
  (FILES (CAR (NLSETQ (DIRECTORY (TCPFTP.SERVER.MERGE.PATHNAMES PATH DEFAULT.PATH T)
  (COND
    (FTPDEBUGFLG (printout FTPDEBUGLOG PATH T)))
    (DISCARDLINE (fetch (TCPFTPCON TCPIN) of TCPFTPCON))
    (TCPFTP.SERVER.RESPONSE 150 (CONCAT "Opening data connection for directory of " PATH " ["
      (LENGTH FILES)
      " file name(s)"]
      (fetch (TCPFTPCON TCPOUT) of TCPFTPCON))
    (LET ((DATASTREAM (TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTPCON USERPORT)))

```

```

(COND
  (DATASTREAM (for FILE in FILES do (PRINT [COND
    (TCPFTP.SERVER.USE.TOPS20.SYNTAX
      (REPACKFILENAME.STRING (PACKFILENAME.STRING
        'HOST NIL 'BODY FILE)
        'TOPS-20))
    (T (TCPFTP.SERVER.RETURN.FILE FILE DEFAULT.PATH
      'INFO]
      DATASTREAM)
      (TERPRI DATASTREAM)
      finally (TCPFTP.SERVER.CLOSE.DATA.CONNECTION TCPFTP.CON))
    (TCPFTP.SERVER.RESPONSE 226 "Data transfer complete" (fetch (TCPFTP.CON TCPOUT)
      of TCPFTP.CON])

```

(TCPFTP.SERVER.EXIT

```

[LAMBDA (CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM) (* ejs%: "20-Mar-86 19:52")
  (CLOSE? CONTROL.OUTPUT.STREAM)
  (CLOSE? CONTROL.INPUT.STREAM])

```

(TCPFTP.SERVER.IDLE.INFO

```

[LAMBDA (PROCESS BUTTON) (* ejs%: "21-Mar-86 16:58")
  (PROMPTPRINT "Idle TCPFTP server")]

```

(TCPFTP.SERVER.LIST

```

[LAMBDA (TCPFTP.CON RDTBL USERPORT DEFAULT.PATH COMMAND) ; Edited 13-Sep-90 14:41 by gadener

```

```

  (* * This function parses USER commands)

```

```

(LET* ((PATH (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
  RDTBL))
  [FILES (CAR (NLSETQ (DIRECTORY (TCPFTP.SERVER.MERGE.PATHNAMES PATH DEFAULT.PATH T]
  (NFILES (LENGTH FILES)))
  (COND
    (FTPDEBUGFLG (printout FTPDEBUGLOG PATH T)))
    (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
    (TCPFTP.SERVER.RESPONSE 150 (CONCAT "Opening data connection for directory " DEFAULT.PATH " ["
      NFILES " file name(s)"]")
      (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
  (LET ((DATASTREAM (TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTP.CON USERPORT)))
    (COND
      (DATASTREAM (for FILE in FILES do (PRINT (COND
        (TCPFTP.SERVER.USE.TOPS20.SYNTAX
          (REPACKFILENAME.STRING (PACKFILENAME.STRING
            'HOST NIL 'BODY FILE)
            'TOPS-20))
        (T (TCPFTP.SERVER.RETURN.FILE FILE DEFAULT.PATH
          COMMAND)))
          DATASTREAM)
          (TERPRI DATASTREAM)
          finally (TCPFTP.SERVER.CLOSE.DATA.CONNECTION TCPFTP.CON))
      (TCPFTP.SERVER.RESPONSE 226 "Data transfer complete" (fetch (TCPFTP.CON TCPOUT)
        of TCPFTP.CON])

```

(TCPFTP.SERVER.MERGE.PATHNAMES

```

[LAMBDA (NAME DEFAULT.PATH NODEVICE.IF.LOCAL NOVERSION.IF.LOCAL DIRFLG) ; Edited 13-Sep-90 14:16 by gadener

```

```

(LET* ((NAMEFIELDS (UNPACKFILENAME.STRING NAME NIL DIRFLG))
  (DEFAULTFIELDS (UNPACKFILENAME.STRING DEFAULT.PATH NIL DIRFLG))
  [HOST (OR (LISTGET NAMEFIELDS 'HOST)
    (LISTGET DEFAULTFIELDS 'HOST])
  [HOSTSPECIFIED (NOT (NULL (LISTGET NAMEFIELDS 'HOST)
    DIRECTORY1)
  (PACKFILENAME.STRING 'HOST HOST 'DEVICE [COND
    ((AND NODEVICE.IF.LOCAL (EQ HOST 'DSK))
      NIL)
    ((OR (LISTGET NAMEFIELDS 'DEVICE)
      (COND
        (HOSTSPECIFIED NIL)
        (T (LISTGET DEFAULTFIELDS 'DEVICE]
      'DIRECTORY
      [COND
        ((LISTGET NAMEFIELDS 'DIRECTORY))
        [COND
          [(SETQ DIRECTORY1 (LISTGET NAMEFIELDS 'SUBDIRECTORY]
          [(SETQ DIRECTORY1 (LISTGET NAMEFIELDS 'RELATIVEDIRECTORY]
          (CL:CONCATENATE 'STRING (LISTGET DEFAULTFIELDS 'DIRECTORY)
            ">" DIRECTORY1))
          (HOSTSPECIFIED NIL)
          (T (LISTGET DEFAULTFIELDS 'DIRECTORY]
        'NAME
        (OR (LISTGET NAMEFIELDS 'NAME)
          (LISTGET DEFAULTFIELDS 'NAME))
        'EXTENSION

```

```

(OR (LISTGET NAMEFIELDS 'EXTENSION)
    (LISTGET DEFAULTFIELDS 'EXTENSION))
'VERSION
(COND
  ((AND NOVERSION.IF.LOCAL (EQ HOST 'DSK))
   NIL)
  (T (OR (LISTGET NAMEFIELDS 'VERSION)
          (LISTGET DEFAULTFIELDS 'VERSION])))

```

(TCPFTP.SERVER.MODE

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 15:38")

(* * This function parses USER commands)

```

(LET ((MODE (READ (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                  RDTBL))
      (RESPONSE.STRING)
      (ERRORFLG))
  (COND
    (FTPDEBUGFLG (printout FTPDEBUGLOG MODE T)))
  (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
  (SELECTQ MODE
    (S (SETQ RESPONSE.STRING "Now in stream mode"))
    (PROGN (SETQ RESPONSE.STRING (CONCAT "Unsupported mode - " MODE))
           (SETQ ERRORFLG T)))
  (COND
    (ERRORFLG (TCPFTP.SERVER.RESPONSE 504 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
              NIL)
    (T (TCPFTP.SERVER.RESPONSE 200 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))

```

(TCPFTP.SERVER.OPEN.DATA.CONNECTION

[LAMBDA (TCPFTP.CON USERPORT FORINPUT)

(* ejs%: "11-Apr-86 16:09")

(* * This function handles opening data connections and marking said tcp connections as busy)

```

(bind (TCB _ (fetch (TCPSTREAM TCB) of (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)))
  DATASTREAM for RETRIES from 0 to TCPFTP.SERVER.RETRYCOUNT
  until (SETQ DATASTREAM (TCP.OPEN (COND
    (USERPORT (CAR USERPORT))
    (T (fetch (TCP.CONTROL.BLOCK TCB.DST.HOST) of TCB)))
    (COND
      (USERPORT (CDR USERPORT))
      (T (fetch (TCP.CONTROL.BLOCK TCB.DST.PORT) of TCB)))
    (SUB1 (fetch (TCP.CONTROL.BLOCK TCB.SRC.PORT) of TCB))
    'ACTIVE
    (COND
      (FORINPUT 'INPUT)
      (T 'OUTPUT))
    T))
  finally (RETURN (COND
    (DATASTREAM (replace (TCPFTP.CON DATASTREAM) of TCPFTP.CON with DATASTREAM)
                 (replace (TCPFTP.CON BUSY?) of TCPFTP.CON with (CREATE.EVENT))
                 (* TELNET standard EOL convention on DATASTREAMS)
                 (SETFILEINFO DATASTREAM 'EOL 'CRLF)
                 DATASTREAM)
    (T (TCPFTP.SERVER.RESPONSE 426 "Couldn't open data connection" (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
       NIL]))

```

(TCPFTP.SERVER.PARSE.PORT

[LAMBDA (PSTRING)

; Edited 28-May-87 18:00 by jop

;;; Parse a port string, in the form 'h1,h2,h3,h4,p1,p2' , where the hx are bytes from an internet host address, and the px are bytes from a 16-bit TCP port
 ;;; number

```

(LET ((IPADDRESS (CREATECELL \FIXP))
      (TCP.PORT 0))
  (bind (BYTECOUNTER _ 0)
        (ACCUMULATOR _ 0)
        ERRORFLG for CH instring PSTRING do (COND
          ((EQ CH (CHARCODE %))
           (COND
             ((IGREATERP BYTECOUNTER 3)
              (SETQ TCP.PORT (IPLUS (ITIMES TCP.PORT 256)
                                     ACCUMULATOR)))
             (T (\PUTBASEBYTE IPADDRESS BYTECOUNTER ACCUMULATOR)))
            (SETQ ACCUMULATOR 0)
            (add BYTECOUNTER 1))
          [(AND (ILEQ CH (CHARCODE 9))
                (IGEQ CH (CHARCODE 0)))
           (SETQ ACCUMULATOR (IPLUS (IDIFFERENCE CH (CHARCODE 0))
                                     (ITIMES ACCUMULATOR 10)))
           (COND
             ((IGREATERP ACCUMULATOR 255)

```

```

                                (SETQ ERRORFLG T)
                                (GO $$OUT)
                                (T (SETQ ERRORFLG T)
                                   (GO $$OUT)))
finally (COND
          (ERRORFLG (RETURN NIL))
          (T (COND
              ((NEQ BYTECOUNTER 5)
               (RETURN NIL))
              (T (RETURN (CONS IPADDRESS (IPLUS (ITIMES TCPPORT 256)
                                                ACCUMULATOR]))))

```

(TCPFTP.SERVER.PASSWORD

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 11:39")

(* * This function parses USER commands)

```

(LET ((PASS (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                      RDTBL)))
  (COND
   (FTPDEBUGFLG (printout FTPDEBUGLOG PASS T)))
  (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
  (TCPFTP.SERVER.RESPONSE 230 "OK, so you're logged in. Now what?" (fetch (TCPFTP.CON TCPOUT)
                                                                           of TCPFTP.CON]))

```

(TCPFTP.SERVER.PATH

[LAMBDA (TCPFTP.CON COMMAND.RDTBL OLDPATH)

; Edited 13-Sep-90 13:37 by gadener

```

(LET* ((PATH (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                      COMMAND.RDTBL))
      (TRUEPATH (TCPFTP.SERVER.MERGE.PATHNAMES PATH OLDPATH NIL T 'RETURN)))
  ;; The last argument, RETURN, makes sure that even though a directory was specified as /a/b/c, we really meant /a/b/c
  )
  (IF TRUEPATH
      THEN (TCPFTP.SERVER.RESPONSE 250 (CONCAT "Default pathname now " TRUEPATH)
                                       (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
      ELSE (TCPFTP.SERVER.RESPONSE 501 (CONCAT "Couldn't interpret " NEWPATH " as a pathname")
                                       (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
      NIL])

```

(TCPFTP.SERVER.PORT

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 11:41")

```

(LET* ((PORTSTRING (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                             RDTBL))
      (PARSEDPORT (TCPFTP.SERVER.PARSE.PORT PORTSTRING)))
  (COND
   (FTPDEBUGFLG (printout FTPDEBUGLOG PORTSTRING T)))
  (COND
   (PARSEDPORT (TCPFTP.SERVER.RESPONSE 200 (CONCAT "User port now " PORTSTRING)
                                           (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
    PARSEDPORT)
   (T (TCPFTP.SERVER.RESPONSE 501 (CONCAT "Couldn't parse port specification " PORTSTRING)
                               (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
    NIL])

```

(TCPFTP.SERVER.PROCESS

[LAMBDA (PORT DEFAULT.FILE.PATH)

; Edited 24-Aug-87 17:55 by scp

(* * This is the TCP-based FTP server top-level)

```

(PROCESSPROP (THIS.PROCESS)
 'INFOHOOK
 (FUNCTION TCPFTP.SERVER.IDLE.INFO))
(LET* ((CONTROL.INPUT.STREAM (TCP.OPEN NIL NIL (OR PORT \TCP.FTP.PORT)
                                             'PASSIVE
                                             'INPUT))
      (CONTROL.OUTPUT.STREAM (TCP.OTHER.STREAM CONTROL.INPUT.STREAM)))
  (* EOL convention -> TELNET Standard)
  (SETFILEINFO CONTROL.OUTPUT.STREAM 'EOL 'CRLF)
  (* Say hello quickly)
  (TCPFTP.SERVER.RESPONSE 220 TCPFTP.SERVER.HERALD.STRING CONTROL.OUTPUT.STREAM)
  (* Spawn a new server)
  (ADD.PROCESS (LIST (FUNCTION TCPFTP.SERVER)
                    PORT
                    (KWOTE DEFAULT.FILE.PATH))
               'RESTARTABLE
               'HARDRESET)
  (* Now that we're "established," errors are fatal)
  (PROCESSPROP (THIS.PROCESS)
 'RESTARTABLE
 'NO)
  (PROCESSPROP (THIS.PROCESS)
 'INFOHOOK
 (FUNCTION TCPFTP.SERVER.CONNECTED.INFO))

```

```

(RESETSAVE NIL (LIST [FUNCTION (LAMBDA (CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM)
                                (COND
                                  (RESETSTATE (TCPFTP.SERVER.ABORTED CONTROL.INPUT.STREAM
                                          CONTROL.OUTPUT.STREAM))
                                  (T (TCPFTP.SERVER.EXIT CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM)
                                     CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM))
                                ]
(TCPFTP.SERVER.COMMAND.LOOP CONTROL.INPUT.STREAM CONTROL.OUTPUT.STREAM DEFAULT.FILE.PATH])

```

(TCPFTP.SERVER.RENAME.FROM

```
[LAMBDA (TCPFTP.CON RDTBL DEFAULT.PATH) (* ejs%: "24-Mar-86 14:16")
```

```
(* * This function parses RNFR commands)
```

```

(LET* [(FILENAME (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                                RDTBL))
        (PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))
        (TRUENAME (CAR (NLSETQ (INFILE PACKED.FILENAME)
                                (COND
                                  (FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
                                  (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
                                ))
        (COND
          (TRUENAME (TCPFTP.SERVER.RESPONSE 350 (CONCAT "About to rename "
                                                         (COND
                                                           (TCPFTP.SERVER.USE.TOPS20.SYNTAX
                                                            (REPACKFILENAME.STRING (PACKFILENAME.STRING
                                                                 'HOST NIL 'BODY TRUENAME
                                                                 'TOPS-20))
                                                           (T TRUENAME)))
                                                         (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
          (T (TCPFTP.SERVER.RESPONSE 550 (CONCAT "File not found - " PACKED.FILENAME)
              (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
          NIL])

```

(TCPFTP.SERVER.RENAME.TO

```
[LAMBDA (TCPFTP.CON RDTBL DEFAULT.PATH FROM.FILE) (* ejs%: "24-Mar-86 14:34")
```

```
(* * This function parses RNTD commands)
```

```

(LET* [(FILENAME (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                                RDTBL))
        (PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))
        (TRUENAME (CAR (NLSETQ (OUTFILE PACKED.FILENAME)
                                (COND
                                  (FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
                                  (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
                                ))
        (COND
          (TRUENAME (RENAMEFILE FROM.FILE TRUENAME)
                    (TCPFTP.SERVER.RESPONSE 250 (CONCAT "Renamed " (COND
                                                                (TCPFTP.SERVER.USE.TOPS20.SYNTAX
                                                                 (REPACKFILENAME.STRING
                                                                  (PACKFILENAME.STRING 'HOST NIL
                                                                  'BODY FROM.FILE)
                                                                  'TOPS-20))
                                                                (T FROM.FILE))
                    " to "
                    (COND
                      (TCPFTP.SERVER.USE.TOPS20.SYNTAX
                       (REPACKFILENAME.STRING (PACKFILENAME.STRING
                                                'HOST NIL 'BODY TRUENAME)
                                                'TOPS-20))
                      (T TRUENAME)))
                    (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON)))
          (T (TCPFTP.SERVER.RESPONSE 553 (CONCAT "Couldn't make an output file named " PACKED.FILENAME)
              (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON])

```

(TCPFTP.SERVER.RESPONSE

```
[LAMBDA (CODE STRING STREAM) (* edited%: "21-Mar-86 11:44")
```

```

(RESETFORM (RADIX 10)
(COND
  (FTPDEBUGFLG (printout FTPDEBUGLOG "> " CODE " " STRING T)))
(printout STREAM CODE " " STRING T))
(FORCEOUTPUT STREAM])

```

(TCPFTP.SERVER.RETRIEVE

```
[LAMBDA (TCPFTP.CON RDTBL USERPORT TYPE DEFAULT.PATH) ; Edited 13-Sep-90 14:59 by gadener
```

```
(* * This function parses USER commands)
```

```

(LET* [(FILENAME (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                                RDTBL))
        (PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))

```

```

(TRUENAME (CAR (NLSETQ (INFILEP PACKED.FILENAME)
(COND
  (FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
(DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
(COND
  [TRUENAME (LET [(FILESTREAM (CAR (NLSETQ (OPENSTREAM TRUENAME 'INPUT 'OLD `((TYPE %, TYPE)
    (COND
      [FILESTREAM (TCPFTP.SERVER.RESPONSE
        150
        (CONCAT "Opening " TYPE " data connection for "
          (COND
            (TCPFTP.SERVER.USE.TOPS20.SYNTAX (REPACKFILENAME.STRING
              (FULLNAME FILESTREAM)
              'TOPS-20))
            (T (FULLNAME FILESTREAM)))
          " ("
          [\IP.ADDRESS.TO.STRING
            (OR (CAR USERPORT)
              (fetch (TCP.CONTROL.BLOCK TCB.DST.HOST)
                of (fetch (TCPSTREAM TCB) of (fetch (TCPFTP.CON TCPIN)
                  of TCPFTP.CON))
              " "
              [OR (CDR USERPORT)
                (fetch (TCP.CONTROL.BLOCK TCB.DST.PORT)
                  of (fetch (TCPSTREAM TCB) of (fetch (TCPFTP.CON TCPIN)
                    of TCPFTP.CON))
                " ) ("
                (OR (GETFILEINFO FILESTREAM 'LENGTH)
                  0)
                " bytes).")
              (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
          (LET ((DATASTREAM (TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTP.CON USERPORT
            )))
            (COND
              (DATASTREAM (LET [(RESULT (NLSETQ (COND
                ((EQ TYPE 'BINARY)
                  (COPYBYTES FILESTREAM
                    DATASTREAM))
                (T (COPYCHARS FILESTREAM
                    DATASTREAM)
                  (CLOSE? FILESTREAM)
                  (TCPFTP.SERVER.CLOSE.DATA.CONNECTION TCPFTP.CON)
                  (COND
                    (RESULT (TCPFTP.SERVER.RESPONSE
                      226 "Data transfer complete"
                      (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON
                        )))
                    (T (TCPFTP.SERVER.RESPONSE
                      426 "Couldn't complete retrieve
                        operation" (fetch (TCPFTP.CON TCPOUT)
                          of TCPFTP.CON))
                      (T (TCPFTP.SERVER.RESPONSE 450 (CONCAT "Couldn't open file " TRUENAME)
                        (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
                      (T (TCPFTP.SERVER.RESPONSE 550 (CONCAT "File not found - " PACKED.FILENAME)
                        (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))

```

(TCPFTP.SERVER.RETURN.FILE

[LAMBDA (FILE DEFAULT.PATH COMMAND)

; Edited 31-Aug-90 17:57 by gadener

;; If COMMAND is LIST , it will return a verbose listing of the file and some of its properties. If the command is NLIST, it will just return the filename
 ;; with extension and version.

;; Note that since the D-mahines don't have a true directory structure, it will return the relative pathname to the file , in relation to DEFAULT.PATH.

```

(LET* [(DEFAULT.PATH.STRING.LENGTH (NCHARS DEFAULT.PATH))
  (PATH (SUBSTRING FILE (PLUS 1 DEFAULT.PATH.STRING.LENGTH)))
  (TAB (CHARACTER (CHARCODE TAB)
    (COND
      ((EQUAL COMMAND 'LIST)
        (CONCAT (GETFILEINFO FILE 'TYPE)
          TAB
          (GETFILEINFO FILE 'LENGTH)
          TAB
          (GETFILEINFO FILE 'WRITEDATE)
          TAB
          (GETFILEINFO FILE 'AUTHOR)
          TAB PATH))
      (T PATH))

```

(TCPFTP.SERVER.STORE

[LAMBDA (TCPFTP.CON RDTBL USERPORT TYPE DEFAULT.PATH)

(* ejs%: "24-Mar-86 15:27")

(* * This function parses USER commands)

```

(LET* [(FILENAME (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
  RDTBL))
  (PACKED.FILENAME (TCPFTP.SERVER.MERGE.PATHNAMES FILENAME DEFAULT.PATH T))

```



```

(TRUENAME (CAR (NLSETQ (OUTFILEP PACKED.FILENAME]
(COND
  (FTPDEBUGFLG (printout FTPDEBUGLOG FILENAME T)))
(DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
(COND
  [TRUENAME (LET [(FILESTREAM (CAR (NLSETQ (OPENSTREAM TRUENAME 'OUTPUT 'NEW
                                              `((TYPE %, TYPE]
              (COND
                [FILESTREAM (TCPFTP.SERVER.RESPONSE 150 (CONCAT "Opening data connection for
                                                              store of " (FULLNAME FILESTREAM)
                                                              )
                  (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
                (LET ((DATASTREAM (TCPFTP.SERVER.OPEN.DATA.CONNECTION TCPFTP.CON USERPORT
                                T)))
                  (COND
                    (DATASTREAM [replace (STREAM ENDOFSTREAMOP) of DATASTREAM
                                         with (FUNCTION (LAMBDA (STREAM)
                                                         (ERROR!])
                    (RESETLST
                     (RESETSAVE (COND
                               ((EQ TYPE 'BINARY)
                                (COPYBYTES DATASTREAM FILESTREAM))
                               (T (COPYCHARS DATASTREAM FILESTREAM)))
                     (LIST [FUNCTION (LAMBDA (FILESTREAM TCPFTP.CON)
                                           (CLOSEF? FILESTREAM)
                                           (TCPFTP.SERVER.CLOSE.DATA.CONNECTION
                                            TCPFTP.CON)
                                           (TCPFTP.SERVER.RESPONSE
                                            226 "Data transfer complete"
                                            (fetch (TCPFTP.CON TCPOUT)
                                                  of TCPFTP.CON)
                                           FILESTREAM TCPFTP.CON))])
                    (T (TCPFTP.SERVER.RESPONSE 450 (CONCAT "Couldn't open file " TRUENAME)
                  (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON]
                (T (TCPFTP.SERVER.RESPONSE 550 (CONCAT "Unable to create output filename - " PACKED.FILENAME)
                  (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON])

```

(TCPFTP.SERVER.STRUCTURE

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 14:08")

(* * This function parses USER commands)

```

(LET ((STRUCTURE (READ (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                        RDTBL))
      (RESPONSE.STRING)
      (ERRORFLG))
(COND
  (FTPDEBUGFLG (printout FTPDEBUGLOG STRUCTURE T)))
(DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
(SELECTQ STRUCTURE
  (F (SETQ RESPONSE.STRING "Now in stream mode")))
  (PROGN (SETQ RESPONSE.STRING (CONCAT "Unsupported mode - " STRUCTURE))
    (SETQ ERRORFLG T)))
(COND
  (ERRORFLG (TCPFTP.SERVER.RESPONSE 504 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
    NIL)
  (T (TCPFTP.SERVER.RESPONSE 200 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON])

```

(TCPFTP.SERVER.TYPE

[LAMBDA (TCPFTP.CON RDTBL)

(* ejs%: "24-Mar-86 15:26")

(* * This function parses USER commands)

```

(LET* ((MAJOR.TYPE (READ (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                          RDTBL))
       (MINOR.TYPE (LET [(TERM.CHAR (BIN (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
          (COND
            ((EQ TERM.CHAR (CHARCODE SPACE))
             (READ (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
                  RDTBL))
            (T (SELECTQ MAJOR.TYPE
                       (A 'N)
                       (L 8)
                       NIL]
              (RESPONSE.STRING)
              (ERRORFLG))
          (COND
            (FTPDEBUGFLG (printout FTPDEBUGLOG MAJOR.TYPE " " MINOR.TYPE T)))
          (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
          (SELECTQ MAJOR.TYPE
            (A (SELECTQ MINOR.TYPE
                       (N (SETQ RESPONSE.STRING "Type is now standard ASCII"))
                       (PROGN (SETQ RESPONSE.STRING (CONCAT "ASCII subtype " MINOR.TYPE " not recognized"))
                             (SETQ ERRORFLG T))))

```

```

(E (SETQ RESPONSE.STRING "EBCDIC not supported")
  (SETQ ERRORFLG T))
(I (SETQ RESPONSE.STRING "Type is now 8-bit binary"))
(L (COND
   ((NEQ MINOR.TYPE 8)
    (SETQ RESPONSE.STRING (CONCAT "Binary byte size " MINOR.TYPE " not supported"))
    (SETQ ERRORFLG T))
   (T (SETQ RESPONSE.STRING "Type is now 8-bit binary"))))
NIL)
(COND
 (ERRORFLG (TCPFTP.SERVER.RESPONSE 504 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
  NIL)
 (T (TCPFTP.SERVER.RESPONSE 200 RESPONSE.STRING (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON))
  (SELECTQ MAJOR.TYPE
   (A 'TEXT)
   'BINARY]))

```

(TCPFTP.SERVER.USER

[LAMBDA (TCPFTP.CON RDTBL)

(* edited%: "21-Mar-86 11:39")

(* * This function parses USER commands)

```

(LET ((USER (RSTRING (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON)
  RDTBL)))
 (COND
  (FTPDEBUGFLG (printout FTPDEBUGLOG USER T)))
 (DISCARDLINE (fetch (TCPFTP.CON TCPIN) of TCPFTP.CON))
 (TCPFTP.SERVER.RESPONSE 230 "Hi, there!" (fetch (TCPFTP.CON TCPOUT) of TCPFTP.CON]))

```

(TCPFTP.SERVER.VERBOSE.LIST

[LAMBDA (FILE STREAM)

(* edited%: "26-Mar-86 11:32")

```

(printout STREAM (COND
  (TCPFTP.SERVER.USE.TOPS20.SYNTAX (REPACKFILENAME.STRING (PACKFILENAME.STRING
    'HOST NIL 'BODY (FULLNAME FILE)
    )
    'TOPS-20))
  (T (FULLNAME FILE)))
";P775252;AFORYOURSELF,"
(FOLDHI (OR (GETFILEINFO FILE 'SIZE)
  0)
  4)
", "
(GETFILEINFO FILE 'CREATIONDATE)
", "
(GETFILEINFO FILE 'WRITEDATE)
T])

```

(TCPFTP.SERVER.WAIT.FOR.IDLE

[LAMBDA (TCPFTP.CON)

(* ejs%: "20-Mar-86 16:39")

```

(bind BUSY? while (SETQ BUSY? (fetch (TCPFTP.CON BUSY?) of TCPFTP.CON)) do (AWAIT.EVENT BUSY?))

```

(TCPFTP.UNIX.LS.DATE

[LAMBDA (FILE)

(* edited%: "21-Mar-86 13:38")

```

(LET* [(CREATIONDATE (GETFILEINFO FILE 'CREATIONDATE))
  (MONTHPOS (STRPOS "-" CREATIONDATE))
  (YEARPOS (STRPOS "-" CREATIONDATE (ADD1 MONTHPOS)))
  (TIMEPOS (ADD1 (STRPOS " " CREATIONDATE)
  (CONCAT (SUBSTRING CREATIONDATE (ADD1 MONTHPOS)
    (SUB1 YEARPOS))
    " "
    (SUBSTRING CREATIONDATE 1 (SUB1 MONTHPOS))
    " "
    (SUBSTRING CREATIONDATE TIMEPOS -4]))

```

)

(RPAQ? TCPFTP.SERVER.HERALD.STRING "Venue Medley FTP Service 1.0 at your service")

(RPAQ? TCPFTP.SERVER.USE.TOPS20.SYNTAX NIL)

(RPAQ? TCPFTP.SERVER.RETRYCOUNT 5)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```

(GLOBALVARS TCPFTP.SERVER.HERALD.STRING TCPFTP.SERVER.USE.TOPS20.SYNTAX TCPFTP.SERVER.RETRYCOUNT)
)

```

```

(FILESLOAD (SYSLOAD)
  TCPFTP)

```

(PUTPROPS TCPFTPSRV COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1990))

FUNCTION INDEX

TCPFTP.SERVER	1	TCPFTP.SERVER.PASSWORD	6
TCPFTP.SERVER.ABORTED	1	TCPFTP.SERVER.PATH	6
TCPFTP.SERVER.ACCOUNT	1	TCPFTP.SERVER.PORT	6
TCPFTP.SERVER.APPEND	1	TCPFTP.SERVER.PROCESS	6
TCPFTP.SERVER.CLOSE.DATA.CONNECTION	2	TCPFTP.SERVER.RENAME.FROM	7
TCPFTP.SERVER.COMMAND.LOOP	2	TCPFTP.SERVER.RENAME.TO	7
TCPFTP.SERVER.CONNECTED.INFO	3	TCPFTP.SERVER.RESPONSE	7
TCPFTP.SERVER.DELETE	3	TCPFTP.SERVER.RETRIEVE	7
TCPFTP.SERVER.DIRECTORY	3	TCPFTP.SERVER.RETURN.FILE	8
TCPFTP.SERVER.EXIT	4	TCPFTP.SERVER.STORE	8
TCPFTP.SERVER.IDLE.INFO	4	TCPFTP.SERVER.STRUCTURE	9
TCPFTP.SERVER.LIST	4	TCPFTP.SERVER.TYPE	9
TCPFTP.SERVER.MERGE.PATHNAMES	4	TCPFTP.SERVER.USER	10
TCPFTP.SERVER.MODE	5	TCPFTP.SERVER.VERBOSE.LIST	10
TCPFTP.SERVER.OPEN.DATA.CONNECTION	5	TCPFTP.SERVER.WAIT.FOR.IDLE	10
TCPFTP.SERVER.PARSE.PORT	5	TCPFTP.UNIX.LS.DATE	10

VARIABLE INDEX

TCPFTP.SERVER.HERALD.STRING	10	TCPFTP.SERVER.RETRYCOUNT	10	TCPFTP.SERVER.USE.TOPS20.SYNTAX	10
-----------------------------	----	--------------------------	----	---------------------------------	----
