

```
-- DebugFTP.Mesa;
-- Edited by:
--       Sandman on April 24, 1978 10:25 AM
--       Barbara on June 20, 1978 9:52 AM
```

#### DIRECTORY

```
AltoFileDefs: FROM "altofiledefs" USING [FP],
CommandDefs: FROM "commanddefs" USING [
  FTPCode, WriteFTPString, WriteIDString],
ControlDefs: FROM "controldefs",
DebugFTPDefs: FROM "debugftpdefs" USING [FetchHandle],
DebugMiscDefs: FROM "debugmiscdefs" USING [
  CommandAbort, confirm, WriteCharZ, WriteEOL],
DebugSymbolDefs: FROM "debugsymboldefs" USING [DCleanSymbolItems],
DebugUtilityDefs: FROM "debugutilitydefs" USING [InvalidateFileCache],
DirectoryDefs: FROM "directorydefs" USING [EnumerateDirectory],
DiskKDDefs: FROM "diskkdefs" USING [CountFreeDiskPages],
IODefs: FROM "iodefs" USING [
  CR, NUL, ReadChar, ReadID, Rubout, WriteChar, WriteDecimal, WriteString],
SegmentDefs: FROM "segmentdefs" USING [
  DefaultAccess, DestroyFile, FileError, InsertFile],
StringDefs: FROM "stringdefs" USING [AppendString, EquivalentString],
SystemDefs: FROM "systemdefs" USING [
  AllocateHeapNode, AllocateHeapString, FreeHeapString];
```

```
DebugFTP: PROGRAM
IMPORTS CommandDefs, DebugMiscDefs, DebugSymbolDefs, DebugUtilityDefs,
DirectoryDefs, DiskKDDefs, IODefs, SegmentDefs, StringDefs, SystemDefs
EXPORTS DebugFTPDefs =
```

#### BEGIN

```
CR: CHARACTER = IODefs.CR;
NUL: CHARACTER = IODefs.NUL;
```

```
FTPData: TYPE = RECORD [
  fetch: DebugFTPDefs.FetchHandle,
  hostname, directoryname, filename: STRING];
```

```
ftp: POINTER TO FTPData ← NIL;
```

```
AddFetch: PUBLIC PROCEDURE[f: DebugFTPDefs.FetchHandle] =
BEGIN
  IF ftp = NIL THEN
    BEGIN OPEN SystemDefs;
      ftp ← AllocateHeapNode[SIZE[FTPData]];
      ftp.hostname ← AllocateHeapString[40];
      ftp.directoryname ← AllocateHeapString[40];
      ftp.filename ← AllocateHeapString[80];
    END;
  ftp.fetch ← f;
  RETURN
END;
```

```
CallFTP: PUBLIC PROCEDURE =
BEGIN OPEN DebugMiscDefs, DebugFTPDefs, IODefs, CommandDefs;
  c: CHARACTER;
  DO
    WriteEOL[];
    WriteString["->"L];
    c ← ReadChar[];
    SELECT UpperCase[c] FROM
      'O =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
          BEGIN
            GetFTPID[open, NUL, ftp.hostname | CommandAbort => LOOP];
            GetFTPID[dir, CR, ftp.directoryname | CommandAbort => LOOP];
            ftp.fetch.ConnectToHost[ftp.hostname, ftp.directoryname];
          END;
        'R =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
          BEGIN
            GetFTPID[retrieve, CR, ftp.filename | CommandAbort => LOOP];
            ftp.fetch.RetrieveFile[ftp.filename];
```

```

    DebugUtilityDefs.InvalidateFileCache[];
    DebugSymbolDefs.DCleanSymbolItems[];
END;
'S =>
IF ftp = NIL THEN WriteFTPString[notinstalled]
ELSE
BEGIN
    GetFTPID[store, CR, ftp.filename | CommandAbort => LOOP];
    ftp.fetch.StoreFile[ftp.filename];
END;
'C =>
IF ftp = NIL THEN WriteFTPString[notinstalled]
ELSE
BEGIN
    WriteFTPString[close];
    confirm[! Rubout => LOOP];
    ftp.fetch.CloseConnection[];
END;
'F =>
BEGIN
    WriteFTPString[freepgs];
    WriteDecimal[DiskKDDefs.CountFreeDiskPages[]];
    WriteFTPString[pgs];
END;
'D =>
BEGIN
    IODefs.WriteChar['D'];
    SELECT UpperCase[ReadChar[]] FROM
    'U =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
        BEGIN
            GetFTPID[dump, CR, ftp.filename | CommandAbort => LOOP];
            ftp.fetch.DumpFiles[ftp.filename];
        END;
    'E =>
        BEGIN
            name: STRING ← IF ftp = NIL
                THEN SystemDefs.AllocateHeapString[40] ELSE ftp.filename;
            GetFTPID[delete, CR, name |
                CommandAbort =>
                BEGIN
                    IF ftp = NIL THEN SystemDefs.FreeHeapString[name];
                    LOOP
                END];
            confirm[! Rubout => LOOP];
            DeleteFile[name];
            IF ftp = NIL THEN SystemDefs.FreeHeapString[name];
            END;
        ENDCASE => WriteFTPString[ques];
    END;
'L =>
BEGIN
    IODefs.WriteChar['L'];
    SELECT UpperCase[ReadChar[]] FROM
    'I =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
        BEGIN
            GetFTPID[list, CR, ftp.filename | CommandAbort => LOOP];
            ftp.fetch.ListFiles[ftp.filename];
        END;
    'O =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
        BEGIN
            GetFTPID[load, CR, ftp.filename | CommandAbort => LOOP];
            ftp.fetch.LoadFiles[ftp.filename];
            DebugUtilityDefs.InvalidateFileCache[];
            DebugSymbolDefs.DCleanSymbolItems[];
        END;
        ENDCASE => WriteFTPString[ques];
    END;
'Q =>
BEGIN
    WriteFTPString[quit];

```

```

        confirm[! Rubout => LOOP];
        EXIT;
        END;
    'X =>
        IF ftp = NIL THEN WriteFTPString[notinstalled]
        ELSE
            BEGIN
                WriteFTPString[chat];
                confirm[! Rubout => LOOP];
                ftp.fetch.Chat[];
            END;
        ENDCASE => WriteFTPString[ques];
    ENDLLOOP;
    IF ftp # NIL THEN ftp.fetch.CloseConnection[];
    RETURN
    END;

GetFTPID: PROCEDURE [command: CommandDefs.FTPCode, c: CHARACTER, s: STRING] =
    BEGIN OPEN DebugMiscDefs, IODefs;
        t: STRING ← [80];
        CommandDefs.WriteFTPString[command];
        StringDefs.AppendString[t, s];
        ReadID[t | Rubout =>
            BEGIN CommandDefs.WriteIDString[del]; SIGNAL CommandAbort; END];
        WriteCharZ[c];
        s.length ← 0;
        StringDefs.AppendString[s, t];
    END;

FP: TYPE = AltoFileDefs.FP;

DeleteFile: PUBLIC PROCEDURE [file: STRING] =
    BEGIN OPEN SegmentDefs, StringDefs;
        fileFound: BOOLEAN ← FALSE;
        FileObj: FP;
        GetFiles: PROCEDURE [fp: POINTER TO FP, name: STRING] RETURNS [BOOLEAN] =
            BEGIN
                IF name[name.length-1] = '.' AND file[file.length-1] # '.'
                    THEN name.length ← name.length - 1;
                IF EquivalentString[name, file] THEN
                    BEGIN FileObj ← fp; fileFound ← TRUE; END;
                RETURN[fileFound];
            END;

        DirectoryDefs.EnumerateDirectory[GetFiles];
        IODefs.WriteString[file];
        IF NOT fileFound THEN CommandDefs.WriteFTPString[notfound]
        ELSE
            BEGIN
                DestroyFile[InsertFile[@FileObj, DefaultAccess]
                    | FileError => GOTO notallowed];
                CommandDefs.WriteFTPString[deleted];
                DebugUtilityDefs.InvalidateFileCache[];
                EXITS
                notallowed => CommandDefs.WriteFTPString[notdeleted];
            END;
        RETURN;
    END;

UpperCase: PROCEDURE [c: CHARACTER] RETURNS [CHARACTER] =
    BEGIN
        RETURN[IF c IN ['a..'z] THEN LOOPHOLE[c + ('A - 'a)] ELSE c];
    END;

END...

```