HP ARPA File Transfer Protocol User's Guide

HP 3000 MPE/iX Computer Systems

Edition 6

HEWLETT® PACKARD

Manufacturing Part Number: 36957-90159
E0300

U.S.A. March 2000

Notice

Restricted Rights Legend

Acknowledgments

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Contents

6. MPE/iX File System and Data Transfers

7. Troubleshooting

A. MPE/iX and POSIX Differences

B. Error Messages

Tables

Preface

The HP ARPA File Transfer Protocol product provides the ARPA file transfer protocol (FTP) service to MPE/iX users.

Related Documentation

The following manuals contain information related to the product described in this manual.

FTP

- *Installing and Managing HP ARPA File Transfer Protocol Network Manager's Guide*

Link Products

- *HP 3000/iX Network Planning and Configuration Guide*
- *Using the OpenView DTC Manager*
- (For NMMGR): *Using the Node Management Services (NMS) Utilities*

General Networking

- *NS 3000/iX Operations and Maintenance Reference Manual*
- *NS 3000/iX Screens Reference Manual*
- *NS 3000/iX Error Messages Reference Manual*

MPE/iX

- *MPE/iX Commands Reference Manual*
- *MPE/iX Error Messages Manual Volume I*, *Volume II*, and *Volume III*

Network Services

- *Using NS 3000/iX Network Services*

Software Installation

- *HP 3000 MPE/iX System Software Maintenance Manual*
- *MPE/iX Documentation Guide*

1 Overview

This chapter briefly describes ARPA services and what is provided on MPE/iX. In addition, who should use this manual is included.

The ARPA Services

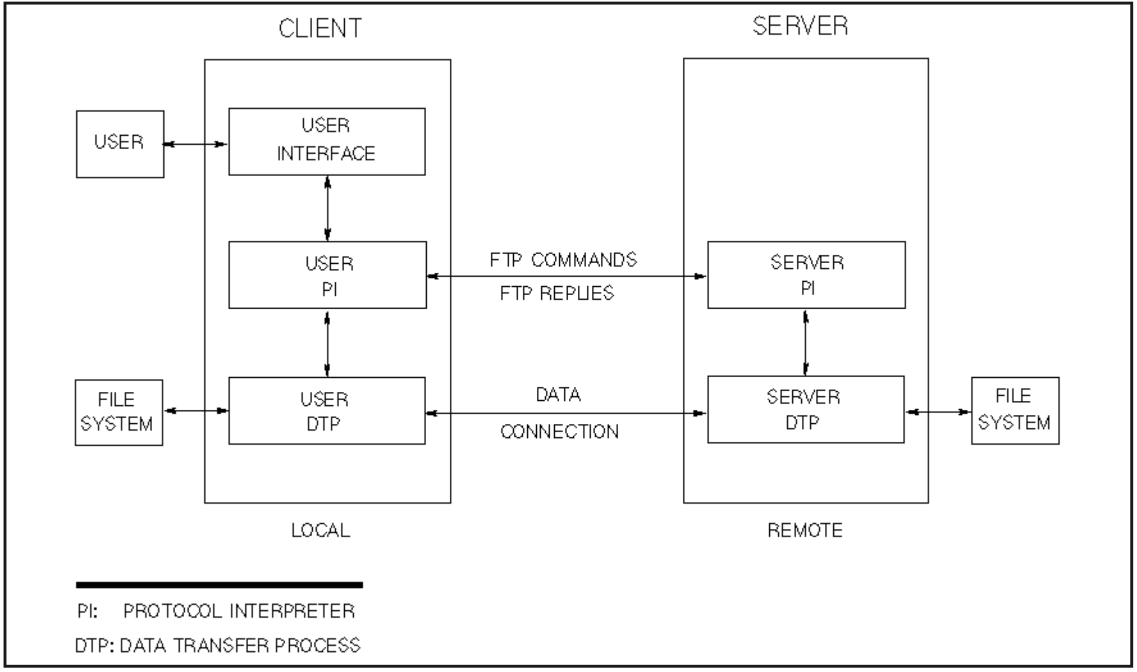
The ARPA services are a subset of the networking services originally developed by the University of California at Berkeley for the Advanced Research Projects Agency (ARPA). ARPA services have become a de facto standard for multivendor network communication.

FTP

FTP is an ARPA Service that allows users to transfer files among other networked systems. FTP is the file transfer program that uses the ARPA standard File Transfer Protocol (FTP). With FTP, you can also perform file management operations.

FTP is based on the client-server model as shown in Figure 1-1.. An FTP user on a system (the client) accesses the remote host FTP server to perform FTP tasks. FTP conforms to the de-facto industry standards RFC959 and RFC1123.

Figure 1-1. FTP Client-Server Model



MPE/iX File Transfer Protocol

The MPE/iX FTP is an implementation of both the client and server programs which conform to RFC959 and RFC1123. The MPE/iX FTP can transfer files with FTP implementations on many other system types, including MS-DOS, OS2, HP-UX, (AT&T, IBM & SUN) variations of UNIX, POSIX, VAX/VMS, IBM/MVS, HP-RTE and other MPE systems.

FTP/iX is supported over Internet Protocol (IP) based network and operates over many different hardware links. These hardware links include, Local Area Network links (LAN): 100VG-AnyLAN, 100Base-T, IEEE 802.3, ETHERNET, IEEE 802.5 (Token Ring), FDDI, and Wide Area Network links (WAN): X.25.

FTP User (Client)

The MPE/iX FTP user can perform the following tasks:

- Connect to a remote host system
- Transfer or delete single or multiple files
- Rename single files on the remote system
- List or change remote directories
- Perform ASCII, binary, or bytestream file transfers
- Change file attributes (using `BUILD` parameters) for files transferred to the MPE/iX system

The supported FTP user commands are summarized in Chapter 2 , “FTP User Commands: Quick Reference.”

FTP Server

Remote users accessing the MPE/iX FTP server can perform the following tasks:

- Establish a session on a remote MPE/iX host system
- Transfer or delete single or multiple files
- Rename single files on the remote MPE/iX system
- List or change remote MPE/iX hierarchical directories
- Perform ASCII, binary, or bytestream file transfers
- Change file attributes (using `BUILD` parameters) for files transferred to the MPE/iX system

The supported FTP Server commands available to remote users are described in Chapter 5 , “Using the FTP Server.”

Who Should Use This Manual

This manual is intended for MPE/iX users who want to transfer files between MPE/iX and other HP and non-HP systems that support the FTP service.

FTP User

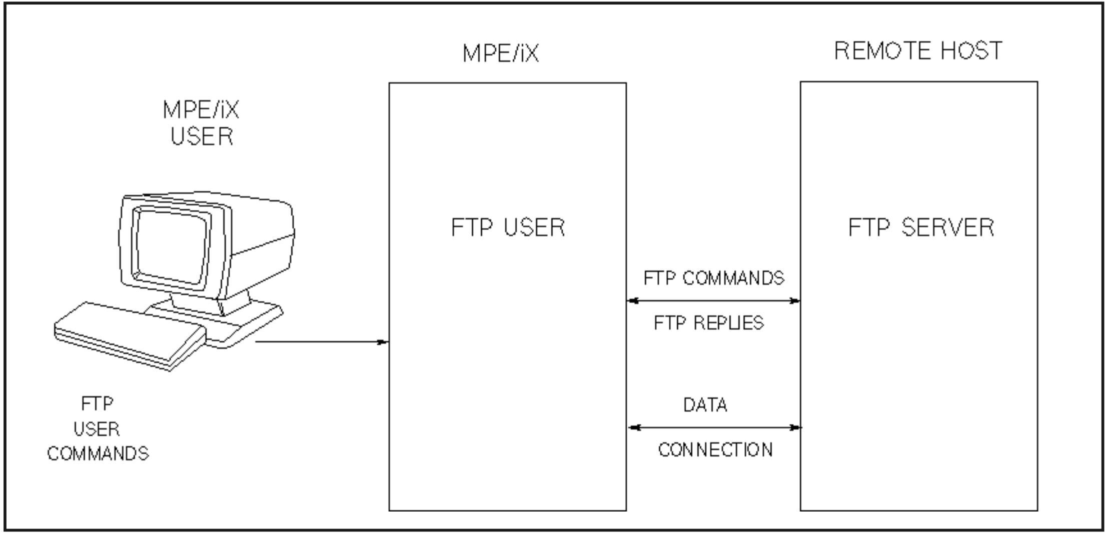
An MPE/iX user can access a remote systems FTP server by running the MPE/iX FTP client program.

The MPE/iX user enters commands to the FTP client program, which sends them to the remote systems FTP server. The remote systems FTP server responds with messages and data, as shown in Figure 1-2..

If you are using FTP for the first time, review the FTP user commands in Chapter 2 , "FTP User Commands: Quick Reference," and Chapter 3 , "Getting Started: A Simple FTP Session," and follow the Simple Session example.

To perform specific FTP tasks such as changing remote directories, or transferring files, see Chapter 4 , "Using FTP," for a detailed task-oriented approach to using FTP.

Figure 1-2. MPE/iX FTP User

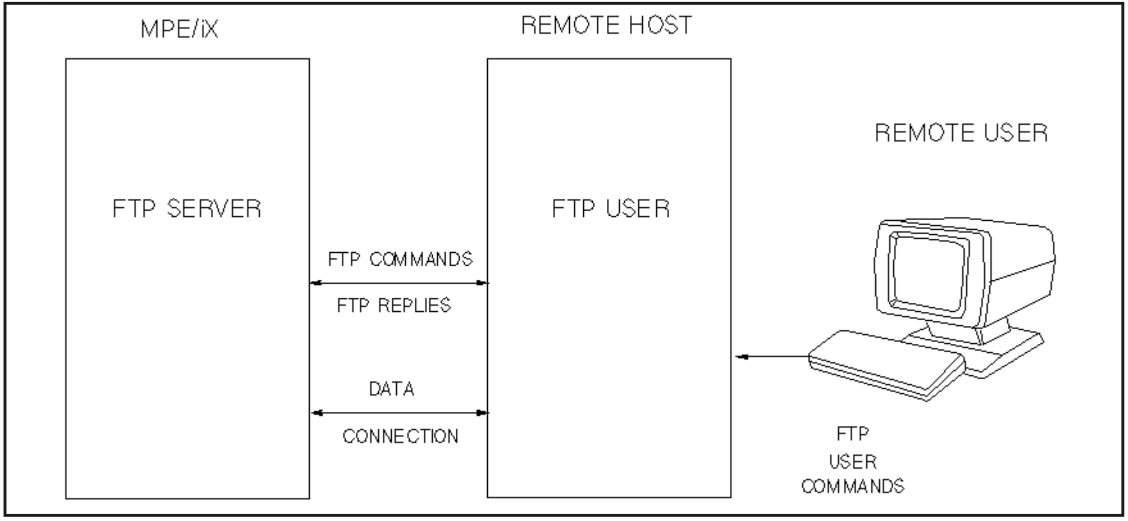


FTP Server

A remote user running an FTP client can access the MPE/iX FTP server as shown in Figure 1-3..

Chapter 5 , "Using the FTP Server," provides information for remote users accessing the MPE/iX FTP server.

Figure 1-3. MPE/iX FTP Server



Before Using the FTP Client

In order to use the FTP client, your network administrator must have a network up and running. The remote host you wish to access must be reachable from your network. You should have some understanding of the remote file system you want to access.

To use the FTP client you will need the following from your network administrator for each system you are accessing:

- The ARPA domain name, remote internet protocol (IP) address, or NS node name.
- User logon identification and passwords.

2 FTP User Commands: Quick Reference

This chapter provides a quick reference of the syntax and usage of the FTP user commands. The commands are listed in alphabetic order.

FTP user commands can be entered in uppercase or lowercase letters.

To stop a file transfer or directory listing in progress, you can use **[CTRL]-Y**. You are prompted to exit FTP, cancel the transfer, or continue. From the FTP prompt you can exit by entering `QUIT` or pressing the **[Break]** key.

:mpecommand

From within an FTP session you can execute an MPE/iX command or program by entering a colon (:) followed by the command or program name. After the command is executed, or the program is exited, you are returned to your FTP session.

? [subject]

A synonym for the `HELP` command. This command will display an informative message about a particular subject. If no argument is given, FTP displays a list of the known commands.

APPEND *localfile* [*remotefile*] [*;buildparms*]

Store data from localfile on the local system into remotefile on the remote system. If remotefile is not specified, localfile will be used as the remote file name as well. If the remote file exists, the data will be appended to it; otherwise, the file will be created.

NOTE If the remote file does not exist, this command behaves the same as `PUT`, and any specified build parameters (buildparms) will be used. But if the file already exists, any buildparms will be ignored.

ASCII

Sets the FTP file transfer type to `ASCII`. This is the default when you first enter the FTP program.

BINARY

Sets the FTP file transfer type to binary.

BYE

Synonym for the `QUIT` command.

BYTESTREAM (BYTE)

Sets the file transfer type to bytestream.

CASE

When using the MPUT command, turns on or off the feature to force resulting filenames to lowercase. When case is off, resulting file names are all uppercase. The default is off.

CD *remotedirectory*

Changes the working directory on the remote host to the directory specified.

CLOSE

ClosesU the remote connection but remains in the FTP program.

DEBUG

Turns on or off the terminal screen display of the commands and parameters that are sent to the remote host. The default is off.

DELETE *remotefilename*

Deletes the remote file specified. A directory path may be included with *remotefilename*.

DIR [*remotedirectory* [*localfile*]]

Writes the requested *remotedirectory* listing to the terminal, or to a local file if specified. If *remotedirectory* is not specified, the current remote working directory contents are listed.

DISCONNECT

A synonym for the CLOSE command.

EXIT

A synonym for the QUIT command.

EXITONERROR

Turns on or off the feature to exit an FTP session when an error is detected. The last error and message are saved in system variables (enter SHOWVAR FTP@ to display). Useful for batch jobs using FTP commands. The default is off.

FORM [NONPRINT]

Sets the FTP file transfer form to the specified format. The only supported form is NONPRINT. Entering FORM without a parameter displays the supported format.

GET *remotefile* [*localfile*] [*;buildparms*] [*;move*]

Transfers the remote file specified to a local file. Build parameters may be used to change how the file is stored on MPE/iX. If a local file name is not specified, the file is saved with the same name as the remote file name (assuming a legal file name).

The `move` option requests that the specified source file be physically moved from the source location to the specified target location.

This is available for `MGET`, `PUT` and `MPUT` as well.

NOTE The removal of the original source file may be guarded by file system security.

HASH

Toggles the state of the hash function within the ftp client. When the hash function is enabled, the “#” (hash) symbol will print to stdlist for every 1024 characters transferred during data transfer requests.

HELP [*ftpcommand*]

Displays a list of the valid FTP user commands. If an FTP user command is specified with `HELP`, a brief description of the command is displayed. You can also use the `HELP` command for information on specific FTP messages and errors, and the supported build parameters (`buildparms`).

LCD *localdirectory*

Changes the working directory on the local machine to the directory specified.

LS [*remotedirectory* [*localfile*]]

Displays the contents of a remote directory, one file per line. If a *localfile* name is specified, the listing will be saved to the file specified. If *remotedirectory* is not specified, the current remote working directory contents are listed.

MDELETE *remotefiles*

Delete multiple files on the remote system.

MGET *remotefiles* [*;buildparms*] [*;move*]

Retrieve multiple files from the remote system. Build parameters can be specified and apply to all files in the transfer. The resulting local files will have the same name as the remote files. Note that MPE/iX file names are exclusively uppercase.

MKDIR *remotedirectory*

Creates a directory on the remote host machine.

MODE [STREAM]

Sets the FTP file transfer mode to the specified mode. The only supported mode is `STREAM`. Entering `MODE` without a parameter displays the supported format.

MODTIME *remotefile*

Returns the last modification time and date of the remote file.

MPUT *localfiles* [*;buildparms*] [*;move*]

Transfer multiple files using wildcard characters in the *localfiles* parameter.

Optionally, build parameters can be specified if the remote system is an MPE/iX system. The build parameters affect each file in the transfer. Non-MPE/iX FTP servers will not interpret the build parameters as intended. The resulting file names are in uppercase letters unless you use the CASE command to force the resulting file names to all lowercase letters.

OPEN *remotehostname*

Opens a connection to the remote host specified. Must be either an ARPA domain name, IP address, or an NS node name.

PROMPT

Toggle interactive prompting. Interactive prompting is used during multiple file transfers to allow the user to selectively retrieve or restore files, and during multiple file deletes to selectively delete. If prompting is turned off, the default for MGET and MPUT is to transfer all files, and the default for MDELETE is to delete all files.

PUT *localfile* [*remotefile*] [*;buildparms*] [*;move*]

Transfers a local file to a file on the remote system. If the remote file name is not specified, the file is saved with the local file name (assuming a legal file name). Optionally, build parameters can be specified if the remote system is an MPE/iX system.

PWD

Displays the name of the remote working directory on the terminal.

QUIT

Closes the connection to the remote system and exits from FTP.

QUOTE *string*

Sends server commands to the remote host. Used for troubleshooting.

RECV *remotefile* [*localfile*] [*:buildparms*]

A synonym for the GET command.

REMOTEHELP [*servercommand*]

Displays the remote host FTP server commands supported. If a server command is specified with the REMOTEHELP command, a brief description of the command is displayed. (Implementation dependent).

RENAME *remotefile newname*

Renames a remote file.

REPLACE

Toggles the internal replace flag and the value of the environment variable FTPREPLACE to either TRUE (on) or FALSE (off). This flag specifies the file placement policy during subsequent file transfers.

`on`	sets the internal replace flag and the value of the FTPREPLACE CI environment variable to TRUE (this is the default state). In this case, all file transfers will unconditionally replace the target file.
`off`	sets the internal replace flag and the value of the environment variable to FALSE. In this state, all file transfers will be conditional predicated on user response to replace the target file if it previously existed.

RESET

Clear reply queue. This command resynchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

RMDIR *remotedirectory*

Deletes a directory from the remote machine.

RUNIQUE

Toggle receive-unique mode. In receive-unique mode, all files stored on the local machine will be guaranteed to be created with a unique file name.

SEND *localfile* [*remotefile*]

A synonym for the PUT command.

SITE *string*

Asks remote server to perform a site-specific command. Type REMOTEHELP SITE for a list of commands supported by the remote server.

SIZE *remotefile*

Returns the size in bytes of the remote file.

STATUS

Display the status of the connection, the data type, whether each of verbose, debug, exit-on-error, prompting, lowercase, store-unique, receive-unique, command tracing and data tracing are on or off, plus the connection timeout value.

STRUCT [FILE]

Sets the FTP file transfer structure to the specified structure. The only supported structure is FILE. Entering STRUCT without a parameter displays the supported structure type.

SUNIQUE

Toggle store-unique mode. In store-unique mode, all files stored on the remote machine will be guaranteed to be created with a unique file name.

SYSTEM

Displays the remote system type.

TIMEOUT [num-secs]

The connection timeout value indicates how long to wait for a message from the remote FTP server before giving up. The allowable range is 0 to 3000. A value from 1 to 3000 indicates a timeout value in seconds. A value of 0 means no timeout (that is, wait forever). If num-secs is not specified, the current timeout value will be displayed. Otherwise, this command sets the connection timeout to num-secs seconds.

TRACE [command | data]

If command is specified, toggle TCP tracing on the command connection; likewise on the data connection if data is specified. If no parameter is given, toggle TCP tracing on both connections.

TYPE [*transfertype*]

Sets the FTP file transfer type to the specified type. If a transfer type is not included, the current file transfer type is displayed. The supported transfer types are `ASCII`, BINARY, and BYTESTREAM (as of 4.5.) The default is `ASCII`.

USER *username*

Used to enter a logon name in order to establish a session on the remote host. A connection to the remote host must already have been established using OPEN.

VERBOSE

Turns on or off the display of responses from the remote host. The default is on.

3 Getting Started: A Simple FTP Session

This chapter provides you with a simple FTP file transfer session. The example shows the output from an MPE/iX user’s terminal accessing an HP 9000. The FTP responses shown are with verbose mode on.

FTP Session Example

To follow the same steps as the example, make sure you have the following:

- Remote logon name and passwords
- Remote system internet protocol (IP) address
- Create or have available a simple text file of type ASCII to transfer to the remote system.

Step 1: Run FTP

After you have entered your MPE/iX user account, enter:

```
:FTP.ARPA.SYS
```

A banner similar to the following is displayed:

```
Transfer Protocol [A0002001] (C) Hewlett-Packard Co. 1995
ftp>
```

Step 2: Connect and Log into the Remote Host

At the ftp> prompt, enter the OPEN command followed by the remote host IP address:

```
ftp> OPEN 123.50.43.21
```

```
220 sysname FTP server (Version $Revision: 15.14 $ $Date: 89/09/23 11:35:11
$) ready.
```

Enter the remote user account and password when prompted:

```
Name (username) myname
331 Password required for myname.
PASSWORD: enter password)
230 User myname logged in.
ftp>
```

Step 3: View Your Local Directory

Enter the MPE/iX command LISTF, preceded by a colon (:) as follows:

```
ftp> :LISTF,2
```

```
ACCOUNT = PUB           GROUP = MYGROUP]
FILENAME   CODE               -----------LOGICAL RECORD-----------
                              SIZE      TYP      EOF      LIMIT
NSCREEN1                      256B      FA       800      10000
NSCREEN2                       80B      FA       500      14000
NSPROGX PROG                 1934W      VB      4551       9480
README                         80B      FA       500      14000
```

Step 4: Transfer a File from MPE/iX to the Remote Host

Select a simple ASCII text file, for example, README from your list of files. Transfer your local file to the remote host file named readme1 using the PUT command as follows:

```
ftp> PUT README readme1
200 PORT command okay.
150 Opening data connection for readme1 (123.50.42.32,50894)
226 Transfer complete
nnn bytes sent in n.nn seconds: (nn.nn Kbytes/sec)
```

Step 5: View the Remote Directory

Enter the DIR command at the ftp>prompt. A listing of the files in the remote working directory is displayed. The file you transferred should be in the list. In this example, readme1 is the third file in the listing.

```
ftp> DIR

200 PORT command okay.

150 Opening data connection for /bin/ls -l (123.50.42.32,50895) (0 bytes).
total 13

-rw-rw-rw-       1 myname     guest      10      Feb 13 18:44      cathy1
-rw-rw-rw-       1 myname     guest     100      Feb 12 18:42      cathy2
-rw-rw-rw-       1 myname     guest     110      Feb 13 18:43      readme1

226 Transfer complete

nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
```

Step 6: Transfer a File from the Remote Host to MPE/iX

From the remote directory listing, choose an ASCII type file to transfer. For this exercise, you could use the file you just transferred and copy it back to your local system with a new name.

Enter the GET command followed by the name of the remote file you want to transfer and then the filename you want to copy into on your local system. For example:

```
ftp> GET readme1 README2

200 port command okay.

150 Opening data connection for readme2 (123.50.42.32,50896) (2997 bytes).

226 Transfer complete.

nnn bytes received in n.nn seconds: (n.nn Kbytes/sec)
```

You can verify that the file transferred to your account by entering a :LISTF command:

```
ftp> :LISTF,2

ACCOUNT = PUB            GROUP = MYGROUP]

FILENAME   CODE               -----------LOGICAL RECORD-----------

                               SIZE       TYP       EOF       LIMIT
NSCREEN1                       256B       FA        800       10000
```

```
NSCREEN2                         80B       FA         500       14000
NSPROGX PROG                   1934W       VB        4551        9480
README                           80B       FA         500       14000
README2                          80B       FA         500       14000
```

Step 7: Exiting the FTP User Program

When you are finished transferring files or performing directory operations, enter QUIT.

```
ftp> QUIT
221 Goodbye.
END OF PROGRAM
:
```

When you issue a QUIT command, you are logged off the remote host, and the remote connection is closed. You are returned to your MPE/iX session prompt.

The Next Step

The previous example showed the steps for a simple file transfer session. The Chapter 4 , “Using FTP,” section, provides more detailed task descriptions of the steps you followed in the simple session, and other features of FTP.

The Chapter 2 , “FTP User Commands: Quick Reference,” section, provides syntax and brief command descriptions for FTP listed in alphabetic order.

4 Using FTP

This chapter provides you with the information you can use to accomplish the following tasks with FTP:

- Running the FTP User Program
- Connecting and Logging on to a Remote Host
- Performing Directory Operations
- Setting or Changing Transfer Types
- Transferring Files
- Renaming Remote Files
- Deleting Remote Files
- Viewing Local Settings
- Issuing MPE/iX Commands from Within FTP
- Temporarily Exiting FTP
- Obtaining Help
- Using FTP Commands in UDCs and Programs
- Obtaining Status Information

NOTE FTP commands can be entered in upper or lowercase letters. Uppercase letters are used in the examples for clarity.

The examples in this chapter show output received for an FTP user on MPE/iX accessing an HP 9000 FTP server.

Running the FTP User (Client) Program

To run the FTP client program from the CI prompt:

```
:RUN FTP.ARPA.SYS [; INFO= "remotehostname"]
```

or

```
:FTP.ARPA.SYS [remotehostname]
```

or on 5.0 and greater, if the HPPATH CI variable has ARPA.SYS in it, you can also run the client program by:

```
:FTP [remotehostname]
```

From the POSIX shell you can run the client program by:

```
shell iX> ftp
```

You can start the FTP user program and connect to a remote host in one step by using the INFO= string in the form:

```
:RUN FTP.ARPA.SYS;INFO="remotehostname"
```

Where *remotehostname* is the ARPA domain name, internet protocol (IP) address, or remote host NS name.

Once the connection has been established, enter the remote user logon name and password when prompted as described in the next section under "Logging On."

Connecting and Logging On to a Remote Host

From FTP, you must establish a connection to the remote host you wish to access.

If you did not enter a remote host name on the command line when you started FTP, you must open a connection to the remote system as follows.

From the ftp> prompt, enter the OPEN command followed by the remote host ARPA domain name, internet protocol (IP) address, or NS node name supplied by your network administrator:

```
ftp> OPEN remotehostname
```

IP Addresses

IP addresses are assigned by your network administrator to uniquely identify computer systems to allow communication between systems on the network.

IP addresses, also called internet addresses, are in the form:

nnn.nnn.nnn.nnn

Where *nnn* is a number from 000 to 255, inclusive. For example, if the IP address of the remote system is 123.1.20.125, you would enter:

```
ftp> OPEN 123.1.20.125
```

Node Names

Besides using IP addresses with the FTP OPEN command, you can use either an ARPA domain name or an NS node name in the OPEN command. NS names are recognized if the remote host supports the Probe protocol or if the remote host name is configured in your local node network directory. ARPA domain names are configured in two ASCII files, RSLVCNF.NET.SYS and HOSTS.NET.SYS. For configuration information, refer to *Installing and Managing HP ARPA File Transfer Protocol Network Manager's Guide*. Obtain the node names from your network administrator. For example, if the name, node2, is configured as a remote host name on your network, you can enter:

```
ftp> OPEN node2
```

For a system identified by a fully-qualified NS nodename such as nodex3.test.hp (in the form: *nodename.domain.organization*, you can enter:

```
ftp> OPEN nodex3.test.hp
```

Logging On

Once the connection has been established, enter the remote user logon name and password when prompted.

You must know the user logon syntax for the remote host you are accessing. For example, an HP 9000 logon name could be: myname. You would enter the following when prompted:

```
Name (username) myname
331 Password required for myname.
PASSWORD:          (enter password)
230 User myname logged in.
```

Passwords are not echoed back to your terminal screen.

Changing Accounts

To log into a different user account on the remote system, use the USER command. After entering the user logon name, you are prompted for any passwords associated with the logon name:

```
ftp> USER myname
331 Password required for myname.
PASSWORD:          (enter password)
230 User myname logged in.
```

Accessing Another Remote Host

After establishing a connection to a remote host, you can access another remote host by entering the CLOSE command, then entering another OPEN *remotehostname* command:

```
ftp> CLOSE
```

```
ftp> OPEN remotehostname
```

Performing Directory Operations

With FTP, you can view the remote working directory, change remote directories, and view or save a listing of the contents of remote directories.

Viewing the Remote Working Directory

The `PWD` command is used to list the name of the remote working directory. The content of the information displayed by `PWD` is dependent on the remote system.

For example, a UNIX implementation could display:

```
ftp> PWD
251 “/users/myname” is the current working directory
```

Viewing the Contents of a Remote Directory

To list the contents of the remote working directory use the `DIR` command. The contents displayed using the `DIR` command are dependent on the remote host directory listing facilities.

In the sample below, the working directory is `mytest`. The first entry is a directory, the remaining entries are files.

```
ftp> DIR

200 PORT command okay.

150 Opening data connection for /bin/ls -l (123.50.42.32,50895) (0 bytes).
total 33

drwxrwxrwx       1 myname     guest      10      Feb 13 18:44      ftest
-rw-rw-rw-       1 myname     guest     100      Feb 12 18:42      file1
-rw-rw-rw-       1 myname     guest     100      Feb 12 18:42      file2
-rw-rw-rw-       1 myname     guest     110      Feb 13 18:43      testt

226 Transfer complete

nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
```

To list the contents of another directory, enter the `DIR` command followed by a valid remote directory name:

```
ftp> DIR remotedirectory
```

Simple directory Listing

The `LS` command provides a consistent display of directory contents for unlike computer systems. One item per line is displayed. For example:

```
ftp> LS
200 PORT command okay.
150 Opening data connection for /bin/ls (123.50.42.32,50896) (0 bytes).
ftest
file1
file2
```

```
testt
226 transfer complete
```

Listing Directory Information to a Local File

You can transfer remote directory information to a local file with the LS or DIR commands using the following form (assuming a remote host UNIX file system):

```
ftp> DIR [remotedirectory/]filename localfile
```

or,

```
ftp> LS [remotedirectory/]filename localfile
```

The format of *remotedirectory/filename* is dependent on the remote system implementation. The *filename* part can contain metacharacters (wildcard characters) supported by the remote system. Some metacharacters supported by the HP 9000 are asterisk (*), and question mark (?).

For example, to transfer a listing of all files in the working directory to your local file REMDIR, enter:

```
ftp> DIR * REMDIR
```

To transfer the contents of the directory named ftest beginning with the letter R to local file REMDIR1, enter:

```
ftp> LS ftest/R* REMDIR1
```

Changing Directories

To change directories in the remote account you are using, enter the CD command as follows:

```
ftp> CD remotedirectory
```

Where *remotedirectory* is the directory you wish to access.

Example

The following example shows a user logged in to an HP 9000 as “myname” who checks the current directory, changes directories, then checks the new current directory.

```
ftp> PWD
251 “/users/myname” is the current working directory.
ftp> CD test
200 CWD command okay.
ftp> PWD
251 “/users/myname/test” is the current working directory.
ftp> CD ..
200 CWD command okay.
ftp> PWD
251 “/users/myname” is the current working directory.
```

Setting or Changing Transfer Types

FTP supports three transfer types, binary, ASCII, and BYTESTREAM. When you first enter FTP, the transfer type defaults to ASCII. For example, text files are type ASCII and are transferred as ASCII files.

To find out what the transfer type is for your current session, enter the `TYPE` command without parameters. For example, if ASCII is the current transfer type, the following displays:

```
ftp> TYPE
Using ASCII mode to transfer file (FTPINFO 25).
```

Binary

To change the file transfer type to binary, enter the command `BINARY` at the `ftp>` prompt as follows:

```
ftp> BINARY
200 Type set to l.
```

You can also use the `TYPE` *`transfertype`* command to change the file transfer type. To set the transfer type to binary, you can enter:

```
ftp> TYPE BINARY
200 Type set to l.
```

ASCII

To change the file transfer type to ASCII, enter the command `ASCII` at the `ftp>` prompt:

```
ftp> ASCII
200 Type set to A.
```

You can also use the `TYPE` *`[transfertype]`* command to change the file transfer type. To set the transfer type to ASCII, you can enter:

```
ftp> TYPE ASCII
200 Type set to A.
```

BYTESTREAM

To change the file transfer type to BYTESTREAM, enter the command `BYTESTREAM` or `BYTE` at the `ftp>` prompt:

```
ftp> BYTESTREAM
200 Type set to L (byte size 8).
```

You can also use the `TYPE` *`[transfertype]`* command to change the file transfer type. To set the transfer type to `BYTESTREAM`, you can enter:

```
ftp> TYPE BYTESTREAM
200 Type set to L (byte size 8).
```

Transferring Files

Once you have logged on to the remote host, you can transfer files from your system to the remote host, or from the remote host to your local system.

Transferring Limitations

The following types of MPE/iX files can be transferred to a remote host using FTP :.

Standard MPE files, with fixed, variable, or bytestream format, and either binary or ASCII data types.

In addition, the latest enhancemnets enable the transfer of the following types of non-standard types between two HP 3000 machines. Both the FTP server and the client must be running the latest version.

1. Circular files
2. Message files
3. RIO files
4. CM KSAM files
5. NM KSAM files
6. HPSPOOL files

Of these, SPOOL files are transferred as fixed ASCII files. The above non-standard file transfers are not possible between an MPE and an non-MPE system. However, SPOOL files can be transferred as fixed ASCII files from an MPE to a non-MPE system.

The following MPE/iX file type cannot be transferred with FTP.

- HFS (Hierarchical File System) directories

In addition, undefined (U) record format files cannot be transferred.

Transferring MPE/iX Files to the Remote Host

Use the PUT command to copy a file on your system to the remote host. FTP does not delete the local copy of the file. The file is saved in the remote working directory.

```
ftp> PUT localfile remotefile
```

The *remotefile* named must follow the file naming conventions for the remote system. If you omit the *remotefile* name, the file will be saved with the *localfile* specified:

```
ftp> PUT localfile
```

If the *localfile* is not a valid file name on the remote system, the file transfer fails.

CAUTION If the remote file name you specify already exists on the remote host, the remote system may overwrite the existing remote file without warning.

File Naming

Note that other systems may support uppercase and lowercase file names. For example, if you entered the following commands and file names when connected to a UNIX system, the result would be two new remote files named RFILE and `rfile`.

```
ftp> PUT RFILE
```

```
ftp> PUT rfile
```

On MPE/iX, RFILE or `rfile` is the same file name.

Transferring Files to a Different Directory

To transfer a file to other than the remote working directory, you can change directories using the CD command, or you can specify the directory name with the *`remotefile`* specified as in the following example (to a UNIX file system):

```
ftp> PUT myfile testdir/myremfile
200 PORT command okay.
150 Opening data connection for testdir/myremfile
226 Transfer complete.
nnn bytes sent in n.nn seconds: (n.nn Kbytes/sec)
```

Transferring Files from Other Groups

The `lcd` command will change directories and consequently the group. For example, you could transfer a file named NSPROG1 from GROUP1 to a remote file named `nsprog` as follows:

```
ftp> PUT NSPROG1.GROUP1 nsprog
```

If the *`remotefile`* is not specified in this example, the remote file name would be:

```
NSPROG1.GROUP1
```

Transferring Multiple Files

To transfer a group of files to the remote system, use the FTP MPUT command.

For example, transfer all files in your account, beginning with the letter “C” to the remote host. First verify that the files are all of one type (ASCII or binary) using the MPE/iX LISTF command:

```
ftp> :LISTF C@,2

ACCOUNT = PUB              GROUP = MYGROUP]

FILENAME   CODE                  -----------LOGICAL RECORD-----------

                                 SIZE      TYP       EOF       LIMIT

CAT1                              80B      FA        850       10000
```

CAT2	80B	FA	800	10000
CAT3	80B	FA	900	10000
CAT4	80B	FA	700	10000

Transfer this group of ASCII files to the remote system:

```
ftp> ASCII
200 Type set to A.
ftp> MPUT C@
mput CAT1? Y
200 PORT command okay.
150 Opening data connection for CAT1 (15.13.132.120,63014).
226 Transfer complete.
nnn bytes sent in n.nn seconds: (n.nn Kbytes/sec)
mput CAT2? Y
```

You are prompted whether or not (yes or no) to transfer each file in the selected group of files.

Pressing **[Enter]** at the prompt defaults to yes.To verify the files transferred:

```
ftp> LS C*
200 PORT command okay.
150 Opening data connection for /bin/ls -1 (123.50.42.32,50895) (0 bytes).
total nn

CAT1
CAT2
CAT3
CAT4
226 Transfer complete
nnn bytes sent in n.nn seconds (n.nn Kbytes/sec
```

Prompting

To transfer this group of files without prompting enabled, use the PROMPT command which turns interactive prompting on and off:

```
ftp> PROMPT
Interactive mode off. (FTPINFO 42)
```

Using Case

MPE/iX only supports uppercase file names. You can use the CASE command to force a group of files transferred with MPUT to lower case as follows:

```
ftp> CASE
Lower case on. (FTPINFO 43)
ftp> MPUT C@
200 PORT command okay.
150 Opening data connection for cat1 (15.13.132.120,63018).
226 Transfer complete.
nnn bytes sent in n.nn seconds: (n.nn Kbytes/sec)
```

NOTE The sequence of commands is repeated for each file transferred.

Check that the files transferred correctly:

```
ftp> LS C@
200 PORT command okay.
150 Opening data connection for /bin/ls -1 (123.50.42.32,50895) (0 bytes).
total nn

Cat1
Cat2
Cat3
Cat4
226 Transfer complete
nnn bytes sent in n.nn seconds (n.nn Kbytes/sec)
```

Transferring Remote Host files to MPE/iX

Use the GET command to copy a file from the remote host to your local MPE/iX system. FTP does not delete the remote copy of the file. Enter the GET command followed by the file name of the remote file, followed by the name of the file you want to copy to on your local system.

```
ftp> GET remotefile localfile
```

If you do not specify the *localfile* name, the file will be transferred with the *remotefile* name specified:

If the *remotefile* name used is not a valid file name on MPE/iX, the file transfer fails.

```
ftp GET remotefile
```

CAUTION If the local file name you specify already exists on your local MPE/iX system, FTP overwrites the existing file without warning.

Default File Attributes

Files copied from the remote host to MPE/iX are saved with default file attributes as follows:

File Type	Record: size (bytes), type	Disk Space
ASCII	80, fixed	204,800
Binary	256, fixed	204,800
BYTESTREAM	1, byte	16,384,000

ASCII 80 Bytes Fixed = Default record length for a terminal screen padded with blanks.

Binary 256 Bytes Fixed = Default record length for a MPE executable program file.

Changing File Building Parameters

You can change how a file is copied to your local system with the MPE/iX BUILD command parameters of the GET or PUT command using the following syntax:

```
ftp> GET remotefile localfile;buildparms
```

The supported BUILD command parameters are:

```
:REC=[-rcsizebytes][,blkfactor[,[{F}][,{BINARY}]]]]
                          {V}   {ASCII}
                          {B}
[;DEV=device]
[;CODE=filecode]
[;DISC=[numrec][,[numextents][,[initialloc]]]]
```

The parameter REC= can also be specified in words (two bytes per word) using a positive integer value. The default *blkfactor* size is one.

HP ARPA FTP supports a subset of the MPE/iX build parameters implemented by the MPE/iX BUILD command. Accordingly, many of the build parameter defaults are overridden by FTP. The FTP build parameter defaults are configured in the ASCII file BLDPARMS.ARPA.SYS and may by modified as desired. Also, users may make local copies and set their own defaults via a file equation like the following.

```
:FILE BLDPARMS.ARPA.SYS=MYFILE
```

See Chapter 6 , “MPE/iX File System and Data Transfers,” for more information about FTP and supported BUILD command parameters. For additional information about BUILD parameters, use MPE/iX online help (HELP BUILD PARMS), see the *MPE/iX Commands Reference Manual*, or read the BLDPARMS.ARPA.SYS file, for more information.

Using file building parameters, you can transfer a file as follows:

```
ftp> GET remfile newfile;REC=-80,,F,ASCII
```

In the next example, using CODE=PROG defaults to a target file of fixed binary, 128 words (which are the REC= default values):

```
ftp> GET filex file2;CODE=PROG
```

Transferring Files to Other Groups

To transfer files to other groups in your account, you can exit from FTP, and re-log on into the other group, or specify the group name as part of the *localfile* specified.

For example, you could transfer a file named nsprog to NSTEST1 in GROUP1 as follows:

```
ftp> GET nsprog NSTEST1.GROUP1
```

Transferring Multiple Files

To transfer a group of files from the remote system to your MPE/iX system, use the `MGET` command.

For example, transfer a group of binary files beginning with the letter “B” to your MPE/iX system. First verify the files to be transferred:

```
ftp> DIR

200 PORT command okay.

150 Opening data connection for /bin/ls -l (123.50.42.32,50895) (0 bytes).
total 33

drwxrwxrwx       1 myname     guest      10      Feb 13 18:44      bprog1

-rw-rw-rw-       1 myname     guest     100      Feb 12 18:42      bprog2

-rw-rw-rw-       1 myname     guest     100      Feb 12 18:42      bprog3

-rw-rw-rw-       1 myname     guest     110      Feb 13 18:43      bprog4

226 Transfer complete

nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
```

Transfer this group of binary files with interactive prompting off:

```
ftp> BINARY
200 Type set to 1.
ftp> MGET b*
200 PORT command okay.
Opening data connection for bprog1 (123.50.42.32,56077) (nnn bytes).
226 Transfer complete.
nn bytes received in n.nn seconds (n.nn Kbytes/sec)
200 PORT command okay.
Opening data connection for bprog2 (123.50.42.32,56077) (nnn bytes).
226 Transfer complete.
nn bytes received in n.nn seconds (n.nn Kbytes/sec)
```

The sequence of commands are repeated for each file transferred.

Check that the files transferred:

```
ftp> :LISTF b@
FILENAME
PROG1  PROG2  PROG3  PROG4
```

Renaming Remote Files

You can rename remote files using the `RENAME` command as follows:

```
ftp> RENAME remotefile newfile
```

To rename a file in other than the working directory on a UNIX-type system, include the directory name with the *`newfile`* parameter as you normally would on the UNIX system as follows:

```
ftp> RENAME remotefile directoryname/newfile
```

CAUTION If the new file name you specify already exists on the remote system, the remote system may overwrite the existing file without warning.

Deleting Remote Files

You can delete files on the remote host from FTP as follows:

```
ftp> DELETE remotefile
```

To delete a file in a different directory on a UNIX-type system, include the directory name with the *remotefile* specified as you normally would on the remote UNIX system as follows:

```
ftp> DELETE directoryname/remotefile
```

Delete Multiple Files

You can delete multiple files on the remote host using the MDELETE command.

With interactive prompting enabled, you are prompted whether or not to delete each file in the fileset you have specified. For example:

```
ftp> MDELETE c*
mdelete cat1? y
200 DELE command okay.
mdelete cat2? y
200 DELE command okay.
mdelete cat3?
200 DELE command okay.
mdelete cata4? n
ftp>
```

Viewing Local Settings

FTP provides commands that determine how files are transferred. For FTP, each of these commands have only one supported option as shown here.

Command	Option
FORM	NONPRINT
MODE	STREAM

```
STRUCT      FILE
```

If you issue one of these commands without the option, the supported option is displayed.

For example if you enter MODE, the following is displayed:

```
ftp> MODE
We only support STREAM mode, sorry. (FTPWARN 18)
```

Issuing MPE/iX Commands from Within FTP

You can enter MPE/iX commands by preceding the command name with a colon (:). The command will execute, and return you to the ftp> prompt on completion. For example you can list your current group files on your local system with the LISTF commands as follows:

```
ftp> :LISTF
```

To list files in other than the current group, include the group name in the LISTF command. For example:

```
ftp> :LISTF @.group1
```

You can also run programs such as an editor from FTP. Upon exiting the program, you are returned to the ftp> prompt.

Temporarily Exiting FTP

From the ftp> prompt, you can temporarily exit FTP without losing the connection to the remote host by using the break key.

Use the MPE/iX RESUME command to return to your FTP session.

Obtaining Help

Use the HELP command for information about the supported FTP user commands. The REMOTEHELP command provides information about the remote host FTP server.

Listing FTP User Commands

Enter the HELP command at the ftp> prompt for a list of FTP supported user commands:

```
 ftp> HELP
 FTP recognizes the following commands:
```

```
 :                dir               mkdir          remotehelp      system
 append           disconnect        mode           rename          timeout
 ascii            exit              modtime        reset           trace
 binary           exitonerror       mput           rmdir           type
 bye              form              open           runique         verbose
 bytestream       get               prompt         send            user
 case             help              put            site            ?
 cd               lcd               pwd            size
 close            ls                quit           status
 debug            mdelete           quote          struct
 delete           mget              recv           sunique

 Additional help available on buildparms, ftperr#, ftpwarn#, ftpinfo#
```

Enter the HELP command with a valid FTP user command to obtain information about individual commands supported. For example:

```
ftp> HELP OPEN
OPEN remote-host
Establish a connection to the specified hot FTP server.
Remote-hot
can either be an ARPA domain name, an MPE/iX NS node name, or an IP address.
```

Listing the Remote FTP Server Commands

The REMOTEHELP command displays the remote host supported server commands. A display similar to the following appears, (depending on the supported server commands).

```
ftp> REMOTEHELP

 214-HP ARPA FTP SERVER recognizes the following commands (*-unimplemented):

 USER     PORT     STOR      MSAM*    RNTO     NLST     MKD      MKD
 PASS     PASV*    APPE      MRSQ*    ABOR*    SITE     XMKD     XMKD
 ACCT*    TYPE     MLFL*     MRCP*    DELE     SYST     RMD      RMD
 SMNT*    STRU     MAIL*     ALLO*    CWD      STAT     XRMD     XRMD
 REIN*    MODE     MSND*     REST*    XCWD     HELP     PWD      PWD
 QUIT     RETR     MSOM*     RNFR     LIST     NOOP     XPWD     XPWD

 214 Refer to FTPDOC.ARPA.SYS for more FTP Server information.
```

Entering REMOTEHELP *command*, where *command* is a valid remote FTP server command, provides you with a brief description of the server command. For example:

```
ftp> REMOTEHELP USER
214 Syntax: USER username
```

FTP server commands may not have the same name as the user commands.

Using FTP Commands in UDCs and Programs

You can use FTP commands in User Defined Commands (UDC), command files and programs to help you perform tasks such as:

- Automating the connection and logon process
- Transferring multiple files
- Creating batch jobs

Using EXITONERROR

The MPE/iX FTP command EXITONERROR allows you to turn on or off the capability to exit an FTP session if an error occurs. The last error and message are saved in CI variables. To show the errors, enter the following at the MPE/iX colon prompt:

```
:SHOWVAR FTP@
```

In the following example, EXITONERROR is activated. The second GET command generates an error because the file does not exist:

```
ftp> EXITONERROR
ExitOnError mode on. (FTPINFO 33)
ftp> GET filex filey
200 PORT command okay.
150 Opening data connection for filey (123.50.42.32,50897) (320 bytes).
```

```
226 Transfer complete.
322 bytes received in 0.65 seconds: (4.25 Kbytes/second)
ftp> GET filez
200 PORT command okay.
550 filez: No such file or directory.
Data Transfer Request Failed. (FTPERR 13)
END OF PROGRAM
```

```
:
:SHOWVAR ftp@
FTPLASTERR = -13
FTPXFERFILES = 0
FTPREQFILES = 0
FTPREPLACE = true
FTPLASTREPLY = 550 filez: No such file or directory
FTPLASTMSG = Data Transfer Request Failed. (FTPERR 13)
:
```

NOTE FTPLASTERR indicates an error if the value is negative and a warning if the value is positive.

Transferring Multiple Files

You can create a command file to automatically copy a group of files from the remote system into your account as in the following example.

1. Create a file with the following contents:

```
parm node, user, pass
continue
purge ftpcmds,temp

echo exitonerror                    > > ftpcmds
echo open !node                     > > ftpcmds
echo open !user !pass               > > ftpcmds
echo get file1 file1                > > ftpcmds
echo get file2 file2                > > ftpcmds
echo get file3 file3                > > ftpcmds
echo quit                           > > ftpcmds

run ftp.arpa.sys;stdin=ftpcmds

if !FTPLASTERR < > 0 then
   showvar FTP@
endif

continue
purge ftpcmds,temp
```

2. From the CI, (:) enter the following to execute the command file you created:

   :yourfile remotehostname username password

   EXITONERROR is enabled in the above example. If there is an error while executing, the error will be printed on the terminal display.

   You can create a UDC using the same set of commands above, except changing the word parm to a UDC name such as getfiles. The first statement for this UDC is:

   getfiles node, user, pass

   Enter the contents of the file in your UDC file. To execute this UDC, you would enter:

   :getfiles *remotehosthame username password*

Using a Jobstream

The following example shows how to use a job file to automate the logon procedure and retrieve a file from a remote system. To execute a job, enter STREAM *jobname* at the MPE/iX prompt (:).

```
!JOB FTPFILE,USERNAME.ACCT,PUB
!RUN FTP.ARPA.SYS
OPEN 123.50.42.32
USER remuser rempass
PWD
DIR*
GET file1
QUIT
```

```
!EOJ
```

Programmatic Example

You can use FTP commands in programs using the HPCICOMMAND intrinsic, as in the following example. The input file containing the FTP commands, FTPIN, is listed following the program.

```
program ftp_prog (input, output);
Const

               msglevel     = 0
               cr           = chr (13);
var
               cmdimage     : string [80]
               cmderror     : shortint;
               parmnum      : shortint;

procedure HPCICOMMAND; intrinsic

begin

{ -----------------------------------------------------    }
{ Run the User-FTP program using the FTPIN input file.     }
{ The FTPOUT file will be created in the temporary space   }
{ and will contain the output of the FTP session.          }
{ -----------------------------------------------------    }

cmdimage := ‘RUN FTP.ARPA.SYS;STDIN=FTPIN;STDLIST=FTPOUT,NEW’ + cr;

               HPCICOMMAND (cmdimage,cmderror,parmnum,msglevel);

               if cmderror < > 0 then

                             begin
                             writelin ( ‘CMDERROR = ‘, cmderror:1 );
                             writelin ( ‘PARMNUM = ‘, parmnum:1 );
                             end:

               end.
```

The contents of the STDIN file FTPIN are listed below:

```
OPEN 123.50.42.32
USER remuser rempass
PWD
DIR*
GET file1
GET file2
GET file3
QUIT
```

Obtaining Status Information

Use the STATUS command to obtain status information regarding the FTP client and server. The status information is useful for debugging purposes and for identifying the states and modes of the client and server peers.

In the following example, both the client (abc.cup.hp.com) and server (katsu.cup.hp.com) are MPE/iX machines.

```
ftp>status
Connected to katsu.cup.hp.com (22.11.138.29). (FTPINFO 40)
Using ASCII mode to transfer files. (FTPINFO 25)
Verbose mode on. (FTPINFO 22)
Debugging off (debug=0). (FTPINFO 23)
ExitOnError mode off. (FTPINFO 34)
Interactive mode on. (FTPINFO 41)
Lower case off. (FTPINFO 44)
Store unique mode off. (FTPINFO 51)
Receive unique mode off. (FTPINFO 53)
Command tracing off. (FTPINFO 55)
Data tracing off. (FTPINFO 57)
Connection time-out value set to 120 seconds. (FTPINFO 49)
211- katsu FTP server status
          Version A0007F06
          Connected to abc.cup.hp.com (25.33.144.142)
          Logged in as ,MANAGER.SYS,PUB
          TYPE: ASCII
          FORMat: Non-print; STRUcture: File; transfer MODE: Stream
          No data connection.
211 End of status.
ftp>
```

5 Using the FTP Server

This chapter provides information for remote users accessing the MPE/iX FTP server and is organized as follows:

- A list of user commands and the corresponding FTP server commands supported.
- Sample session for a remote user accessing the FTP MPE/iX server.
- More information about FTP and MPE/iX for the remote user.

`User command` **(SERVER command)**

DESCRIPTION

`ASCII` **(TYPE A)**

Sets the FTP file transfer type to ASCII. This is the default when you first enter FTP.

`BINARY` **(TYPE I)**

Sets the FTP file transfer type to binary.

`BYTESTREAM` **(TYPE L)**

Sets the FTP file transfer type to bytestream.

`CD` *`remotedirectory`* **(CWD | XCWD)**

Changes the working directory on the remote host to the directory specified.

`CLOSE` **(QUIT)**

Closes the remote connection but remains in the FTP program.

`DELETE` *`remotefilename`* **(DELE)**

Deletes the remote file specified. A directory path may be included with the *`remotefilename`*.

`DIR [`*`remotedirectory`*`][`*`localfile`*`]` **(LIST)**

Writes the requested remote directory listing to the terminal, or to a local file if specified. If *`remotedirectory`* is not included, the current remote working directory contents are listed.

`FORM [NONPRINT]` **(FORM)**

Sets the FTP file transfer form to the specified format. The only supported form is `NONPRINT`.

`GET` *`remotefile`* `[`*`localfile`*`]` **(RETR)**

Transfers the remote file specified to a local file. If a local file name is not specified, the file is saved with the same name as the remote file name (assuming a legal file name).

`LS [`*`remotedirectory`*`][`*`localfile`*`]` **(NLST)**

Displays the contents of a remote directory, one file per line. If a *`localfile`* name is specified, the listing will be saved to the file specified. If *`remotedirectory`* is not included, the current remote working directory contents are listed.

`MDELETE` *`remotefiles`* **(NLST,DELE)**

The remote system expands the wildcard characters in *`remotefiles`* and deletes the set of files on the remote MPE/iX host.

`MGET` *`remotefiles`* **(NLST,RETR)**

The remote system expands the wildcard characters in *`remotefiles`* and retrieves the set of files from the remote host.

`MKDIR` *`remotedirectory`* **(MKD | XMKD)**

Creates a directory on the remote machine.

`MODE [STREAM]` **(MODE)**

Sets the FTP file transfer mode to the specified mode. The only supported mode is STREAM.

`MODTIME` *`remotefile`* **(MDTM)**

Returns the last modification time and date of the remote file.

`MPUT` *`localfiles`* **(NLST,STOR)**

The local system expands the wildcard characters in *`localfile`*s and transfers the set of files from the local host to the remote MPE/iX system.

`OPEN` *`remotehostname`*

Opens a connection to the remote host specified.

`PUT` *`localfile`* `[`*`remotefile`*`] [;`*`buildparms`*`]` **(STOR)**

Transfers a local file to a file on the remote MPE/iX system. Build parameters may be used to change how the file is stored on MPE/iX. If the remote file name is not specified, the file is saved with the local file name provided the local file name is a valid name on the remote MPE/iX system.

`PWD` **(PWD)**

Displays the name of the remote working directory on the terminal. On MPE/iX, the session information is returned as well.

`QUIT` **(QUIT)**

Closes the connection to the remote system and exits from FTP.

`REMOTEHELP [`*`servercommand`*`]` **(HELP)**

Displays the remote host (MPE/iX) FTP server commands supported. If a server command is specified with the REMOTEHELP command, a brief description of the command is displayed.

RENAME *remotefile newname* **(RNFR,RNTO)**

Renames a remote file.

RMDIR *remotedirectory* **(RMD|XRMD)**

Deletes a directory from the remote machine.

```
SITE <sp> HELP
       | MPE/iX
       | BUILDPARMS file-name
       | USER_LABELS  filename [number]
       | STREAM file-name
       | TIMEOUT num-secs
```

(SITE)

Remote system

HELP Returns a help line.

MPE/iX Sets an internal flag indicating that the local system is an MPE/iX machine.

BUILDPARMS Gives the build parameters of the specified file.

USERLABELS Gives the user label listing of the specified file.

STREAM Streams a given job-file.

TIMEOUT Sets the server time-out to the specified value.

SIZE *remotefile* **(SIZE)**

Returns the size in bytes of the remote file.

SRUCT [FILE] **(STRU)**

Sets the FTP file transfer structure to the specified structure. The only supported structure is FILE.

SYSTEM **(SYST)**

Displays the remote system type.

TYPE [*transfertype*] **(TYPE)**

Sets the FTP file transfer type to the specified type. The supported transfer types are ASCII and BINARY and BYTESSTREAM.

USER *sess,user.acct,group* **(USER)**

Used to enter a logon name in order to establish a session on the remote host. A connection to the remote host must already have been established using OPEN. Passwords may be included in the logon string as follows:

```
sess,user/userpass.acct/acctpass,group/grppass
```

Note: Passwords can only be entered once. If you enter passwords with the logon string, *do not* re-enter them at the password prompt, just press the

[Return] key. If you enter passwords as part of the logon string, they are displayed in the message text of the password prompt.

Sample FTP Session: Accessing MPE/iX

The purpose of this sample session is to provide help for a remote user accessing the MPE/iX FTP server. More detailed information about FTP behavior follows the sample.

In order to follow the same steps as the sample on your system, you must obtain the following from your network administrator:

- Internet Protocol (IP) address of the MPE/iX system
- MPE/iX logon account and passwords
- Names of the groups in the logon account containing the files you want to access.

Start FTP

Start FTP from your system.

Connect to the Remote MPE/iX System

To open a connection to MPE/iX use the OPEN command followed by the internet protocol (IP) address for the MPE/iX system. For example:

```
ftp> OPEN 123.50.43.21
220 MPE/iX File Transfer Protocol Server [A0002001] (C) Hewlett-Packard Co.
1990
```

Log on to MPE/iX

Supply the MPE/iX logon account and password when prompted, or use the USER command. For example:

```
ftp> USER sess1,user.myacct,mygroup
331 Password required for SESS1,USER.MYACCT,MYGROUP
[userpass][,accountpass][,grouppass]
Password: userpass,acctpass,grouppass (enter passwords)
230 User logged on.
```

Anonymous FTP

If anonymous FTP is supported on the remote MPE FTP server, the user can login anonymously by giving the username as “anonymous” or “ftp”.

```
ftp> o 15.70.188.48
connected to 15.70.188.48.
220 HP ARPA FTP Server [A0008OZ3] (C) Hewlett-Packard Co. 1990
Name (15.70.188.48:rram): anonymous
```

```
331 Guest login ok, send ident as password.
Password:
230 User logged on
Remote system type is MPE/IX
```

Check the Session and Group

Check the session and group you are accessing. This example shows the user logged on with the working group, MYGROUP.

```
ftp> pwd
257 “/MYACCT/MYGROUP” is the current directory.
257 “SESS1,USER.MYACCT,MYGROUP” is the current session.
```

View Files in the Working Group

Examine the files in the working group using the FTP DIR command:

```
ftp> :DIR *
200 PORT command ok.
150 File LISTF opened; data connection will be opened
ACCOUNT = MYACCT             GROUP = MYGROUP]

FILENAME   CODE               -----------LOGICAL RECORD-----------

                              SIZE     TYP       EOF      LIMIT

NSCREEN1                      256B      FA       800      10000

NSCREEN2                       80B      FA       500      14000

NSPROGX PROG                 1934W      VB      4551       9480

PROGRAMX                      256B      FA       700       1350

256 Transfer complete
nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
ftp>
```

Transfer a File to Your System

Transfer a remote MPE/iX file to your system using the GET command:

```
ftp> GET nscreen1
200 PORT command ok.
150 File NSCREEN1 opened; data connection will be opened
226 Transfer complete
nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
ftp>
```

The file will be saved as nscreen1 on your system.

View Files in a Different Group

Examine the files in another group using the LS command with the group name:

```
ftp> LS @.group1
200 PORT command ok.
```

```
150 File LISTF opened; data connection will be opened
AFILE.GROUP1
TSCREEN.GROUP1
TSCREEN2.GROUP1
NSPROG1.GROUP1
NSPROG2.GROUP1
NSPROG3.GROUP1
NPROGRAM.GROUP1
226 Transfer complete
nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
ftp>
```

Note that this display is different than using the LS command with no group name which displays only file names.

Transfer a File from Another Group

Transfer a binary file from MPE/iX to your system from other than the current working group:

```
ftp> BINARY
200 Type set to l.
ftp> GET NPROGRAM.GROUP1
```

The MPE/iX file will be saved as NPROGRAM.GROUP1 on your system.

Transfer Multiple Files from MPE/iX to Your System

To transfer multiple files from MPE/iX to your system use the MGET command:

```
ftp> MGET @
```

All files in the working group will be transferred and have the same names as on MPE/iX (up to 8 alphanumeric characters, all capitals).

Transfer multiple files from other than the working group on MPE/iX to your system:

```
ftp> MGET N@.GROUP1
```

The resulting file names are:

```
NSPROG1.GROUP1
NSPROG2.GROUP1
NPROGRAM.GROUP1
```

Transfer a File from Your System to MPE/iX

Transfer a local file to the MPE/iX working group using the PUT command.

```
ftp> PUT yfile xfile
```

Use File Building Parameters

Transfer a local file to other than the MPE/iX working group, using MPE/iX file building parameters:

```
ftp> PUT file2 newfile.group1;rec=-78,,f,ascii
```

Transfer Multiple Files to MPE/iX

Transfer multiple ASCII files to MPE/iX using the MPUT command:

```
ftp> ASCII
200 Type set to A.
ftp> MPUT A*
```

If your system supports prompting, you are prompted whether or not (yes or no) to transfer each file in the selected group of files.

Check that the files transferred:

```
ftp> :DIR A@
200 PORT command ok.
150 File LISTF opened; data connection will be opened
ACCOUNT = MYACCT      GROUP= MYGROUP]

FILENAME   CODE                  -----------LOGICAL RECORD---------

                                  SIZE      TYP       EOF       LIMIT

AFILE                            80B       FA        800       204800

AFILE123                         80B       FA        800       204800

AFILE124                         80B       FA        800       204800

AFILENAM                         80B       FA        800       204800

AFILEXX                          80B       FA        800       204800

256 Transfer complete
nnn bytes received in n.nn seconds (n.nn Kbytes/sec)
ftp>
```

If the file names are longer than eight characters, the file transfer fails. Note that all files were saved with the default FTP ASCII file attributes: records of fixed length and 80 bytes.

More Information About FTP and MPE/iX

This section describes, in more detail, MPE/iX logon syntax and file system behavior when using FTP.

MPE/iX User Accounts

MPE/iX user logon accounts are in the form:

```
  ftp> USER sessname,username.acctname,groupname
```

The sessionname parameter (*sessname*) is optional. Passwords may be required for all three parts of the user account:

```
  userpass,accountpass,grouppass
```

You can enter the user logon account and passwords together as follows:

```
  ftp> USER session1,username/userpass.acctname/acctpass,groupname/grouppass
```

Press the return key when prompted for passwords.

The following example shows: no session, and a password for the account only:

```
ftp> USER myname.myacct/acctpass
331 Password required for MYNAME.MYACCT/ACCTPASS
[userpass][,acctpass][,grouppass]
Password:          (press return)
230 USER LOGGED ON
ftp> pwd
257 “/MYACCT/PUB” is the current directory.
257 “,MYNAME,MYACCT,PUB” is the current session
```

NOTE Each password can only be entered once. For example, if you enter the account password as part of the USER string, you *cannot* re-enter it at the password prompt.

If you enter passwords as part of the logon string, they are displayed in the message text of the password prompt.

Groups

An MPE/iX user account can contain many groups, each containing files. A group is comparable to a directory, but only to one level. Unlike UNIX systems, the MPE/iX file system is not hierarchical.

The MPE/iX account manager can assign a default group for a user account. For example, the user account MYNAME.ARPACCT is assigned a default group PUB (with no group password). The user logon is as follows:

```
ftp> USER MYNAME/userpw.ARPACCT/acctpw
331 Password required for MYNAMEUSPERPW.ARPACCT/ACCTPW
[userpass][,acctpass][,grouppass]
Password:          (press return)
230 USER LOGGED ON
ftp> pwd
257 “/ARPACCT/PUB” is the current directory.
257 “,MYNAME,ARPACCT,PUB” is the current session
```

Changing Groups

The FTP CD command is supported on the MPE/iX FTP server as of release 4.5. To change groups you CD as follows:

```
ftp> pwd
257 “/MYACCT/GROUP1” is the current directory.
257 “,MYNAME.MYACCT,GROUP1” is the current session.
ftp> cd ../GROUP2
250 CWD file action successful.
ftp> pwd
257 “/MYACCT/GROUP2” is the current directory.
257 “,MYNAME.MYACCT,GROUP1” is the current session.
```

Notice that only the current directory changed and not the current session.

File Naming on MPE/iX

The MPE/iX file system is not case sensitive: `file1` is the same file as `FILE1`.

A fully-qualified MPE/iX filename is in the form:

```
filename.groupname.account
```

Each part, (*filename*, *groupname*, and *account*) is 1–8 alphanumeric characters, beginning with an alphabetic character. To avoid file naming problems, always explicitly specify resulting filenames in correct MPE/iX format.

Using Metacharacters

Be aware if you use metacharacters (wildcards) and specify a group and/or account with `MGET`, the resulting filenames will be fully qualified MPE/iX file names (*filename.groupname.account*), a maximum of 26 characters, including periods. If your system supports fewer characters than MPE/iX, the filenames may be truncated. Use the `LS` command to verify the set of files you are transferring.

Both * and @ can be used as wildcard characters with the `LS` and `DIR` commands.

Changing File Building Parameters

When transferring files from your system to MPE/iX, you can use file-building parameters following the `PUT` command string:

```
ftp> PUT remotefile localfile;buildparms
```

The supported build parameters are:

```
:REC=[-rcsizebytes][,blkfactor[,[{F}][,{BINARY}]]]]
                                 {V}   {ASCII}
                                 {B}
[;DEV=device]
[;CODE=filecode]
[;DISC=[numrec][,[numextents][,[initialloc]]]]
```

The parameter `REC=` can also be specified in words (two bytes per word) using a positive integer value. The default *blkfactor* size is one. For more information about the `BUILD` command, see the *MPE/iX Commands Reference Manual.*

The default MPE/iX file-transfer specifications for the `PUT` command in the form of file-building parameters are:

```
ASCII: ;REC=-80,,F,ASCII;DISC=204800
Binary: ;REC=-256,,F,BINARY;DISC=204800
BYTESTREAM: ;REC=,,B;DISC=16384000
```

Examples

The following example shows how to transfer a file to MPE/iX with a record size of 150 bytes:

```
ftp> PUT filex file2;REC=-150,,V,ASCII
```

In the following example, using only CODE=PROG defaults to a file of fixed binary, with records of 128 words (which are the BUILD command REC= defaults).

```
ftp> PUT filex file2;CODE=PROG
```

6 MPE/iX File System and Data Transfers

This chapter describes MPE/iX file system characteristics that affect how files are transferred and saved using FTP. This chapter can be useful to both MPE/iX FTP users and remote users accessing the MPE/iX FTP server and is organized as follows:

- How FTP Transfers Files.
- Using BUILD Parameters with FTP
- MPE/iX File System
- File Transfer Examples

How FTP Transfers Files

Files can be transferred with FTP in ASCII mode, binary (IMAGE) mode, and BYTESTREAM mode.

ASCII Transfers

FTP ASCII file transfers are used primarily for the transfer of ASCII text files. The actual data transfer occurs in 8-bit NVT-ASCII representation. The receiver converts the data from the standardized transfer form to the desired target form according to the target system's internal representation requirements. The end-of-record (EOR) for ASCII transfers is represented by a carriage return followed by a line feed.

For the MPE/iX FTP implementation, ASCII data transfers are used when data is transferred in FTP ASCII mode to or from an MPE/iX file of the ASCII file type.

Binary (IMAGE) Transfers

FTP IMAGE type, commonly called binary transfer type is used for efficient storage and retrieval of files and for the transfer of binary data. The data is sent in a contiguous stream of 8-bit transfer bytes. The receiver is required to store the data as contiguous bytes.

For the MPE/iX FTP implementation, binary data transfers are used when data is transferred in FTP binary mode to or from a MPE/iX file of the binary file type. There is no end-of-record designator for binary transfers.

Bytestream (Byte—Local-byte 8) Transfers

FTP bytestream mode sets the file transfer type to a byte size of 8 bits (local-byte 8). This mode is identical to binary, except that inbound files will be created bytestream instead of binary if no build-parms are specified. Outbound files are treated as a binary (Image) type transfer.

NOTE When running FTP on an HP-UX machine, the command to switch to local-byte mode is called "tenex."

MPE/iX Type vs. FTP Transfer Type

Do not confuse MPE/iX file types with FTP file transfer types. Using `BUILD` command parameters, the only MPE/iX file type supported is `STD`

MPE/iX files with an ASCII, BINARY, or BYTESTREAM attribute are specified as part of the `BUILD` command `REC=` parameter.

Using BUILD Parameters with FTP

MPE/iX BUILD command parameters can be used to modify the default file building parameters for files transferred to an MPE/iX system.

When no BUILD command parameters are specified with the target MPE/iX file designation, the MPE/iX FTP server builds the file with the following defaults depending on the transfer mode used:

For ASCII file transfers:

```
REC=-80,,F,ASCII;DISC=204800
```

which translates to a file with 80 byte fixed length ASCII records. The maximum number of records the file can contain is 204,800.

For binary file transfers:

```
REC=-256,,F,BINARY;DISC=204800
```

which translates to a file with fixed length binary records, with a maximum record size of 256 bytes. The maximum number of records the file can contain is 204,800.

For bytestream file transfers:

```
;REC=,,B;DISC=16384000
```

which translates to a file with byte length records, with a maximum record size of 1 byte. The default number of records is 16,384,000.

Build Command Parameters

An MPE/iX FTP user can change how a file is copied from a remote system to MPE/iX by using the MPE/iX BUILD command parameters with the GET command using the following syntax:

```
ftp> GET remotefile localfile;buildparms
```

For a remote user (accessing the MPE/iX FTP server), you can save a file on MPE/iX using the following syntax with the PUT command:

```
ftp> PUT localfile remotefile;buildparms
```

Syntax

The supported BUILD command parameters are:

```
:REC=[-rcsizebytes][,blkfactor[,[{F}][,{BINARY}]]]]
                                {V}    {ASCII}
                                {B}
[;DEV=device]
[;CODE=filecode]
[;DISC=[numrec][,[numextents][,[initialloc]]]]
```

Parameters

REC=-recsizeinbytes

A negative value indicates record size in bytes. A positive integer value indicates record size in words (two bytes per word). If DEV= is not specified, the default record size is 128 words.

blkfactor

The default blkfactor size is one. When creating variable length files, include the 1 in the parameter string to maintain the record size specified in the parameter REC=.

F, V, B

Indicates fixed (F), variable (V), byte (B), length records. The default is fixed (F).

BINARY
ASCII

Indicates the type of records the file contains: BINARY for binary coded records or ASCII for ASCII coded records. BINARY is the default.

DEV=device

Specifies where the file will be stored. The default is to a disk file.

CODE=filecode

A code indicating a specially formatted file. Example: CODE=PROG. Default is none.

DISC=numrec

The value that affects file transfer success is numrec. This value sets the maximum number of records the file can contain. Must be large enough for the target file or the file transfer will fail. The default is 1,023 records.

When you specify BUILD parameters in an FTP transfer, you do not have to specify the MPE/iX file type standard (STD) since it is the default. Note that STD is the only supported MPE/iX file type.

For more information about BUILD parameters, use MPE/iX online help (HELP BUILD PARMS), or see the *MPE/iX Commands Reference Manual*.

Examples Using BUILD Parameters

The first example shows an MPE/iX user transferring a file from the remote system to an MPE/iX system using only the CODE=PROG parameter:

```
ftp> GET filex file2;CODE=PROG
```

The file is created with the BUILD command defaults: fixed binary, 128 word records, with a maximum number of records equal to 1,023.

The second example shows an MPE/iX user transferring a group of binary files from the remote system to an MPE/iX system:

```
FTP> BINARY
200 Type set to l.
ftp> MGET P* ;CODE=PROG
```

All files in the set beginning with letter “P” are created with the following attributes: file code label PROG, fixed binary, 128 word records, and a maximum number of records equal to 1,023. The blank space preceding the semicolon is required.

The third example shows a remote user transferring a file from their system to an MPE/iX system using BUILD parameters as follows:

```
ftp> PUT myfile MPEFILE;Rec=-256,1,F,BINARY;DISC=11000;CODE=PROG
```

An MPE/iX fixed binary file named MPEFILE is created with records of 256 bytes, with a maximum record size of 11,000. The file code label is PROG.

MPE/iX File System

The MPE/iX file system is record oriented. Each file has a file limit (DISC parameter in BUILD) which determines the maximum number of records the file can hold. Each record has a maximum record size which indicates the maximum number of bytes each record can hold. In addition, a file may have fixed or variable length records.

Typically, variable length records are more efficiently stored on the system, and less efficiently accessed. For fixed length record files, storage on the system is less efficient because of byte padding, and access is faster because the records are all the same size.

Fixed Record File Padding

Files created on MPE/iX with fixed records may result in the padding of the records to the record size specified. ASCII files are padded with blank spaces, and binary files are padded with zeros.

For example, a file named vfile containing variable length records is transferred to an MPE/iX file as follows:

```
ftp> ascii
ftp> get vfile fixfile
```

The default file building parameters for ASCII transfers is fixed records of 80 bytes in length. The resulting file will be stored with 80 byte records. Records from vfile that contain fewer bytes of data will be padded with spaces.

Record Truncation

Record truncation can occur when performing a data transfer to an ASCII file on the MPE/iX system. This occurs when the number of bytes in the data transfer exceeds the maximum record size bytes in the target file. When this occurs, the target record is filled to the maximum record size bytes from the data transfer, and the remaining bytes for the current record are discarded. A warning message is displayed when this occurs.

For example, a file named zfile containing 132 byte records is transferred to MPE/iX as follows:

```
ftp> ascii
ftp> get zfile mpefile
Some records were truncated during transfer. (FTPWARN 15)
```

The default ASCII file building parameters are fixed, 80 byte records. Since the original record size is 132 bytes, only the first 80 bytes of each record are transferred, and the remaining bytes in each record are discarded.

End-of-File Limitation

When you transfer a file to an MPE/iX system using FTP, it is important to define enough MPE/iX file records for the resulting MPE/iX file. If you do not use BUILD parameters, the default maximum number of records for both ASCII and binary transfers is 204,800 records. If you are transferring a file that exceeds this default, use the BUILD parameter DISC=numrec to increase the number of records in the target file.

Be careful if you use BUILD parameters. If you do not explicitly specify the DISC=numrec parameter, the default of 1,023 records is used and may not be sufficient.

If the number of records in the target MPE/iX file is insufficient, the target file is not saved, and the existing file (if any) is preserved. An error message is displayed if this occurs.

File Transfer Examples

The following examples show how files are stored and retrieved between an MPE/iX system and a UNIX system using FTP.

Variable Record Files

The first example in Figure 6-1. shows an FTP user on MPE/iX transferring an ASCII file to a UNIX system then retrieving the file with the original record structure intact.

AFILE is stored with the attributes: REC=-80,1,V,ASCII. AFILE can contain records of up to 80 characters. The records are of variable length. The file transfer is performed:

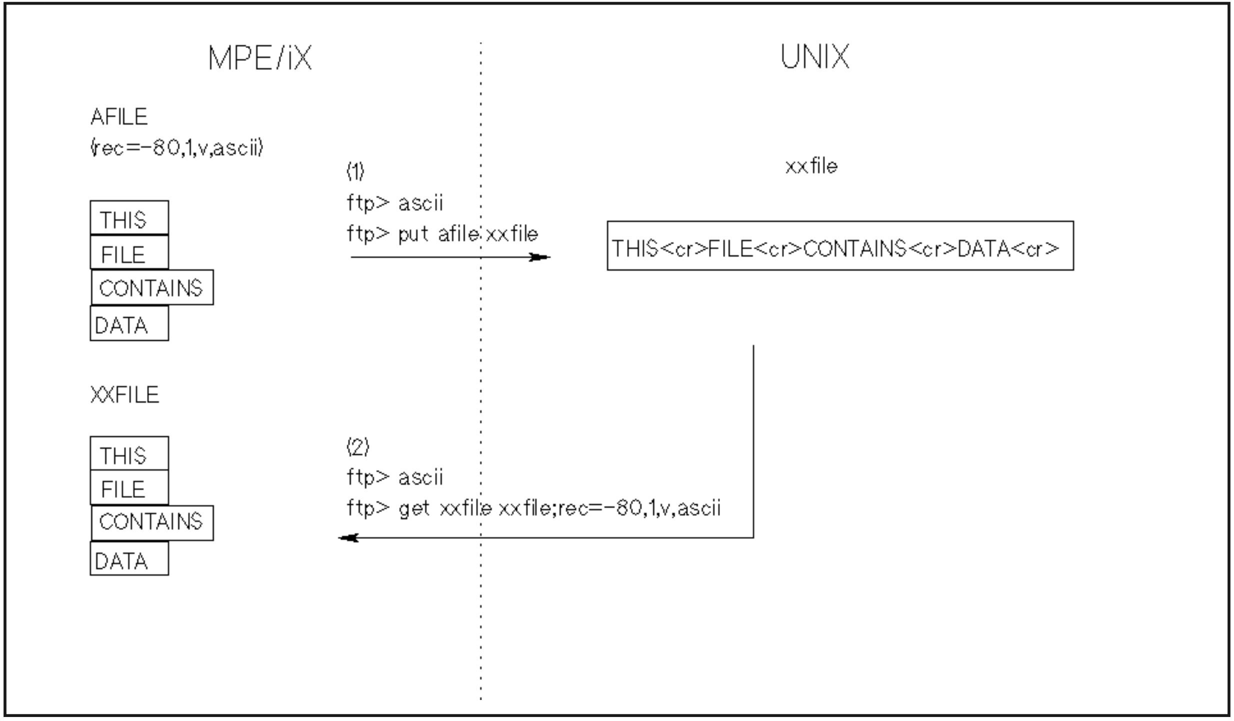
```
(1)
ftp> ascii
ftp> put afile xxfile
```

The UNIX file system stores the data in a contiguous stream, separated by carriage return characters (<cr>).

To retrieve the file as it was originally stored, the MPE/iX FTP user can do the following:

```
(2)
ftp> ascii
ftp> get xxfile xxfile;rec=-80,1,v,ascii
```

Figure 6-1. ASCII Transfer



Mixed Mode Transfer

If you transfer an ASCII type file in binary mode, you may not be able to retrieve the file in its original form as pictured in Figure 6-2., “Mixed Mode Transfer,”.

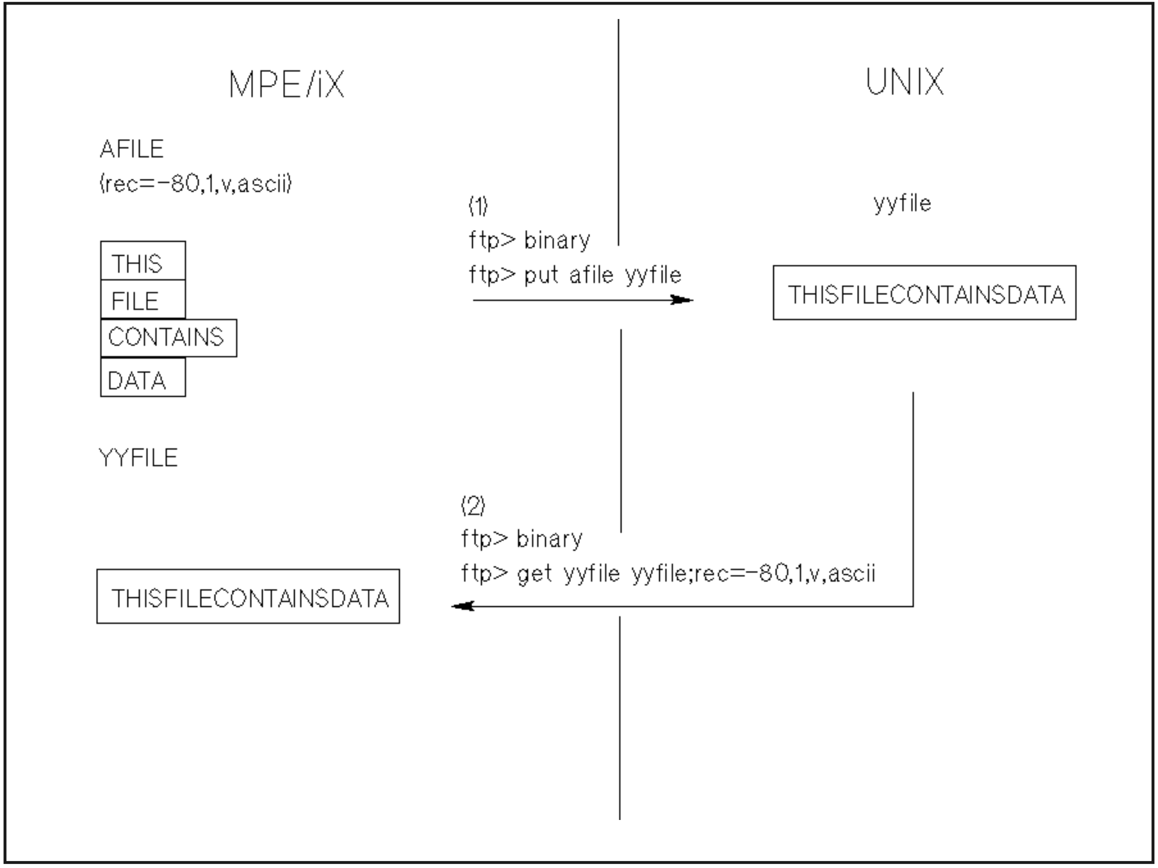
An MPE/iX ASCII file, AFILE, transferred to the UNIX system using FTP binary transfer mode results in a file of contiguous data on the UNIX system. Note that record structure is not maintained in binary transfers.

```
(1)
ftp> binary
ftp> put afle yyfile
```

When the user tries to retrieve the file using file building parameters (`rec=-80,1,v,ascii`), the file is retrieved as one record (since the total amount of data is less than 80 bytes).

```
(2)
ftp> binary
ftp> get yyfile yyfile;rec=-80,1,v,ascii
```

Figure 6-2. Mixed Mode Transfer



Fixed Record Files

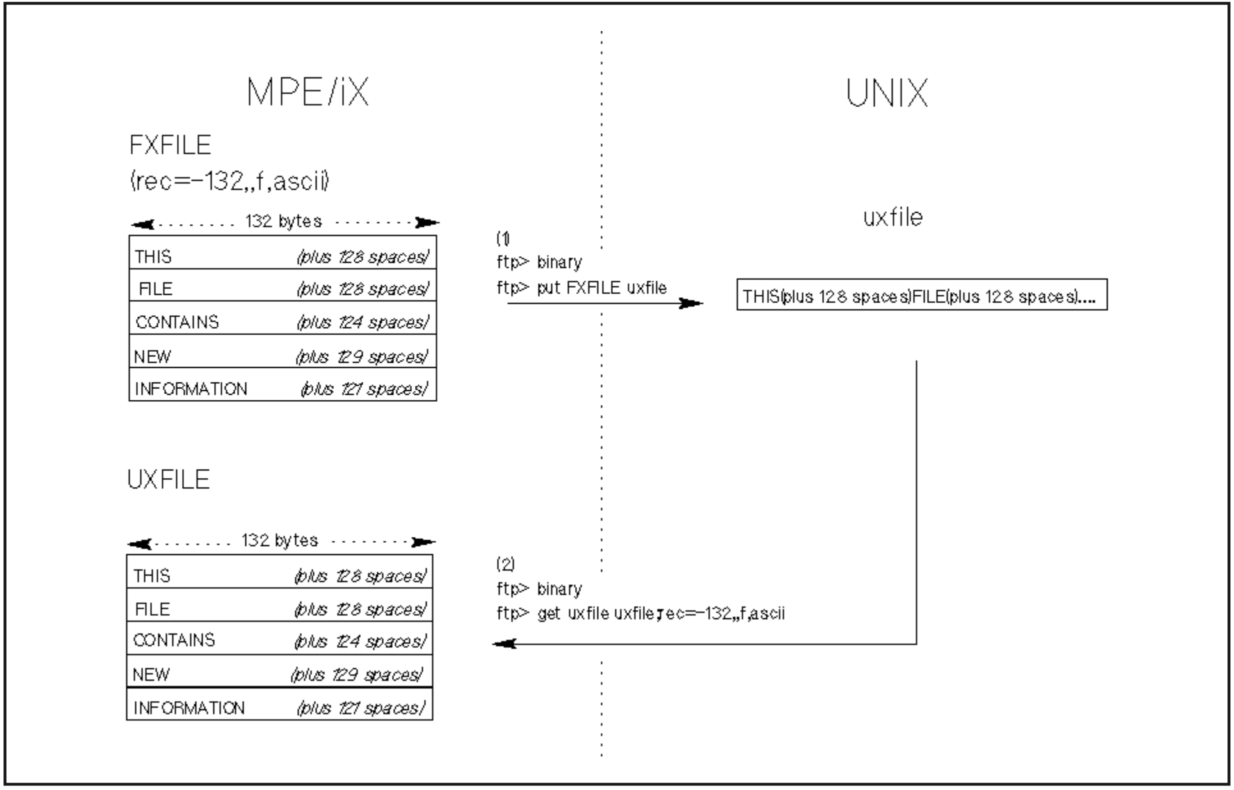
Transferring files that contain imbedded carriage return-line feed characters (`<cr> <lf>`) can be a problem if you try to transfer these files in ASCII mode. In FTP ASCII transfer mode, the remote system interprets a carriage return-line feed pair (`<cr> <lf>`) as a new record. One method of transferring a fixed file to a remote system then retrieving it with the data and record structure intact is pictured in Figure 6-3., "Fixed Record Transfer,". An MPE/iX FTP user transfers an ASCII file with fixed length records of 132 bytes long to a UNIX system in FTP binary mode as follows:

```
(1)
ftp> binary
ftp> put FXFILE uxfile
```

On the UNIX system, the file is stored as a contiguous stream, including the spaces. To retrieve the file and maintain the original record structure, the MPE/iX user must use the GET command with the following build parameters:

```
(2)
ftp> binary
ftp> get uxfile uxfile;rec=-132,,f,ascii
```

Figure 6-3. Fixed Record Transfer



THIS	(plus 128 spaces)
FILE	(plus 128 spaces)
CONTAINS	(plus 124 spaces)
NEW	(plus 129 spaces)
INFORMATION	(plus 121 spaces)

```
(1)
ftp> binary
ftp> put FXFILE uxfile
```

THIS(plus 128 spaces)FILE(plus 128 spaces)....

```
(2)
ftp> binary
ftp> get uxfile uxfile;rec=-132,,f,ascii
```

THIS	(plus 128 spaces)
FILE	(plus 128 spaces)
CONTAINS	(plus 124 spaces)
NEW	(plus 129 spaces)
INFORMATION	(plus 121 spaces)

7 Troubleshooting

This chapter describes possible problems you may encounter and the methods you can use to troubleshoot them.

Cannot Run FTP

- Did you specify the correct *`filename.group.account`* to run the FTP user program:

  `FTP.ARPA.SYS`

- Check with your network administrator: The network and links must be active.

Cannot Connect to the Remote Host

- Did you enter the correct ARPA domain name, IP address, or NS node name? See your network administrator.
- Check with your network administrator to find out the following:
  - — Does the remote host system support FTP?
  - — Is the FTP server up and running on the remote host system?
  - — Is the network link active on the remote host system?
  - — If an NS node name was used, make sure it is correctly entered in the network directory. If an ARPA domain name was used, make sure the name has been configured in the two required ASCII files. If not, use the IP address.

Cannot Logon to Remote (MPE/iX User)

- If an MPE/iX user cannot logon to the remote system, verify the logon account and passwords are correct.
- Try to log on from the remote system with the user's account and passwords.

User Cannot Logon to MPE/iX

- If a remote user accessing MPE/iX cannot logon, verify that the logon account and passwords are correct.
- If you tried to log on from a system that supports prompting, try to log on using the FTP USER command:

  ```
  ftp> USER session,user.account,group
  ```

- Check that passwords are entered correctly. Passwords must be entered as follows:

At the password prompt: `userpass,acctpass,grouppass`

or, as part of the logon string:

```
ftp> USER sess,user/userpass.acct/acctpass,gourp/grouppass
```

NOTE Passwords can only be entered once. For example, if the user enters the account password in the `USER` command string, *do not* re-enter it at the password prompt.

File Transfer Problems

For file transfer problems, first turn verbose and debug on.

- Use the `LS` or `DIR` command to verify the file name spelling.
- Check how file names are entered: some systems support both uppercase and lowercase file names. Some systems support characters in file names that are not supported on MPE/iX.
- On MPE/iX, using `MPUT`, resulting files are in uppercase letters (on non-MPE/iX system). Use `CASE` command to force file names to lowercase.
- Supported file types: must be binary or ASCII, fixed or variable records.
- Check the default file transfer type you are using. Enter:

  ```
  ftp> TYPE
  ```
- Check the type of file you are transferring: Are you transferring binary to ASCII? See Chapter 6 , “MPE/iX File System and Data Transfers.”
- Check the size of records, and file types supported on the remote host and MPE/iX (see Chapter 6 , “MPE/iX File System and Data Transfers,” in this manual for MPE/iX).

Command Not Supported on Remote System

Use the FTP command `REMOTEHELP` to determine the supported server commands on the remote host.

A MPE/iX and POSIX Differences

FTP/iX has the ability to access files in both the MPE/iX and POSIX (IEEE Portable Operating System Interface for Computing Environments) name space. The POSIX standards define a common portable interface to files. The following information defines access and command usage while operating in the POSIX name space environment.

- Filename size restrictions on the MPE/iX file system
- MPE/iX and POSIX file identifiers
- Working with directories
- Listing MPE/iX or POSIX files
- Bytestream files
- Transferring files

Filename Size Restrictions

The file reference can be either in MPE/iX or POSIX namespace.

The MPE file domain consists of MPE accounts, MPE groups and files which reside in these `groups.accounts`. The MPE file domain is a 3 level Hierarchical File System (HFS).

The POSIX file domain is considered a *superset* of the MPE file domain.

The POSIX file domain consists of a root (/) and POSIX directories, which can contain additional directories or files. The POSIX file domain is a multi-level Hierarchical File System.

POSIX is implemented on MPE/iX in such a manor that it is a *superset* of MPE. MPE accounts are now under the POSIX root (/) and are treated as *special* directories; likewise MPE groups are located under accounts and again are treated as *special* directories.

As a result of this implementation, files in the MPE file domain can be accessed by MPE applications; and, since the POSIX file domain is a *superset* of the MPE file domain, POSIX applications can also access these files. In general MPE applications cannot access files in the POSIX file domain unless the application, or the file procedures it calls, are specifically coded to do so.

MPE/iX Namespace

If the file reference does not begin with a dot or a slash, it is parsed according to the MPE/iX syntax and has the following format:

```
filename [/lockword] [.groupname [.acctname]]
```

MPE/iX names must contain from one to eight alphanumeric characters, beginning with an alphabetic character. If *acctname* is specified, you must create directory (CD) access to the target group in the account. The default *groupname* and *acctname* are the logon group and account. Filenames in MPE name space are not case sensitive, and are all shifted to uppercase.

POSIX Namespace

If the file reference begins with a dot (.) or a slash (/), it is parsed according to the POSIX syntax. In this case the file reference can be ./Directory 1/Directory 2/Directory 3/... .

The following syntax rules apply:

- File names are not shifted to uppercase.
- File names can be up to 254 characters in length for absolute pathnames, and 253 characters for relative pathnames.
- File names can begin with, and contain, any of the following characters:

```
        a–z      A–Z,      0–9,      _,      .
```

- File names can contain (but not begin with) a dash (-).
- File names are of the form (`path/filename`) where the path/filename combination can

have a maximum of 255 characters.

For more information, refer to the New Features of MPE/iX: *Using The Hierarchical File System Reference Manual* for accepted syntax of file names and directories.

Sample File Name	MPE File Name Space	POSIX File Name Space
FILENAME	FILENAME	FILENAME
filename	FILENAME	filename
longfilename	Invalid MPE file name	longfilename
Read_Me	Invalid MPE file name	Read_Me

NOTE The file identifiers "./FILENAME" and "./filename" refer to two different files in the POSIX name space since character case is accepted in the POSIX name space.

Example 1:

Files in the MPE file domain in `PUB.SYS` (MPE syntax) are also viewed as being a part of the POSIX file domain in `/SYS/PUB/` (POSIX syntax).

```
:listf log@.pub.sys
FILENAME
LOG       LOG0112      LOG0113
```

or

```
:listfile /SYS/PUB/LOG@
PATH= /SYS/PUB
LOG       LOG0112      LOG0113
```

Example 2:

Files in the POSIX file domain in `"/SYS/PUB/directory1/directory2/"` (POSIX syntax) are outside of the MPE file domain.

```
:listf /SYS/PUB/directory1/directory2/
```

File name missing. (CIERR 531)

```
:listfile /SYS/PUB/directory1/directory2
PATH= /SYS/PUB/directory1/
directory2/
PATH = /SYS/PUB/directory1/directory2/
TESTDATA
```

NOTE The MPE `listf` command executes against files in the MPE file domain. The MPE command `listfile` created for POSIX executes against files in both the MPE and POSIX file domains.

Working With Directories

When a user logs on as USER.ACCT,GROUP, the current working directory is initialized to /ACCT/GROUP. If a user creates a hierarchical directory through the FTP/iX MKDIR DIR1 command and then changes to that directory with the FTP/iX CD command, the new working directory will be /ACCT/GROUP/DIR1. The current working directory can be identified with the FTP/iX command PWD.

To change directories on the local system, use the FTP/iX LCD command. The LCD command will automatically preface (./) to the given name if it does not start with a dot or a slash. This command identifies the new current directory.

To remove directories, use the FTP/iX rmdir command. The (./) is prefaced to the given file name if it does not start with a dot or a slash.

The rmdir will not work on MPE/iX groups or accounts. Likewise, mkdir creates new directories, not groups or accounts.

Note that (.) refers to the current directory, (..) refers to the directory above the current directory, and (/) is used as the separator in the path name. Thus if a file name starts with (.) or (..), the path traversed to find the file is relative to the current directory, whereas if a file name starts with (/), the path is independent of the current directory and refers to the absolute path name.

To change to a different directory outside of the FTP/iX subsystem in the POSIX name space, use the CHDIR command.

Commands

CHDIR

Changes the process's current working directory (CWD) on the local system.

Syntax

```
CHDIR [ [DIR=]dir_name] [;SHOW | NOSHOW]
```

Parameters

dir_name — The name of the new current directory (optional). The dir_name is assumed to be an MPE/iX name. HFS-named directories can be changed if dir_name starts with a dot (.) or a slash (/). The dir_name may not end in a slash and wild carding is not allowed. If dir_name is omitted, your logon directory is used. Your logon directory is your logon group, in the form:

/LOGON_ACCOUNT/LOGON_GROUP

whereas if you need to change to a different MPE/iX group name, use the CHGROUP command.

CHGROUP

Switches you from the current group to another group within the logon account to which you are allowed access (NM capability required). This command is not executable through the FTP command prompt.

Syntax

```
CHGROUP [ [GROUP=]groupname] [/grouppass]
```

Parameters

groupname The name of the group to which you will be switched. If you omit this parameter, you will be switched to the home group.

grouppass The password of the group you are switching to, if it is assigned a password. In a session, if the target group has a password and you fail to supply one on the command line, MPE/iX will prompt you to enter one. You have three tries to enter the correct password before the command fails.

In a batch job, if the target group has a password and you fail to supply one, MPE/iX issues an error message “INCORRECT PASSWORD (CIERR 1441)” and the job fails.

In either case, when you switch to your home group, you may omit the password.

Refer to the *MPE/iX Commands Reference Manual* for command syntax and usage.

Command Line (CI) Examples

Example 1:

```
:showme
USER: #S26,MANAGER.SYS,PUB      (NOT IN BREAK)

:showvar HPCWD
HPCWD = /SYS/PUB
```

Example 2:

```
:showme
USER: #S26,MANAGER.SYS,PUB      (NOT IN BREAK)

:showvar HPCWD
HPCWD = /SYS/PUB

:chgroup work

:showme
USER: #S36,MANAGER.SYS,WORK      (NOT IN BREAK)

:showvar hpcwd
HPCWD = /SYS/WORK
```

Example 3:

```
:showme
USER: #S26,MANAGER.SYS,PUB      (NOT IN BREAK)

:showvar HPCWD
HPCWD = /SYS/PUB
```

```
:chdir /SYS/WORK
CWD is “/SYS/WORK”.

:SHOWME
USER: #S36,MANAGER.SYS,WORK      (NOT IN BREAK)

:showvar hpcwd
HPCWD = /SYS/WORK
```

FTP Usage Examples

Example 1:

This is a sample of the operation of the POSIX current working directory “cwd” and the resulting file domain on the local system. In this example, the user on the local system, logged on as “MANAGER.SYS,PUB”, enters FTP and then executes a “:CHDIR /SYS/WORK” command. The result is the local domain for files, FTP view will be “/SYS/WORK/” (POSIX syntax) or “WORK.SYS” (MPE syntax).

```
:ftp.arpa.sys
File Transfer Protocol [A0006018] (C) Hewlett-Packard Co. 1990

ftp> :showvar HPCWD
HPCWD = /SYS/PUB

ftp> :showme
USER: #S27,MANAGER.SYS,PUB      (NOT IN BREAK)

ftp> :chdir /SYS/WORK
CWD is “/SYS/WORK”.

ftp> :showvar HPCWD
HPCWD = /SYS/WORK

ftp> :showme
USER: #S27,MANAGER.SYS,PUB      (IN PROGRAM)
```

Example 2:

This is a sample of the operation of the POSIX current working directory “cwd” and the resulting file domain on the remote system. In this example, the user connects to the remote system with FTP, logged on as “MANAGER.SYS,PUB” and then executes a FTP “cd /SYS/WORK” command. The result is the remote domain for files FTP will view will be “/SYS/WORK” (POSIX syntax) or “WORK.SYS” (MPE syntax).

```
ftp> pwd
257-“/SYS/PUB” is the current directory.
257 “,MANAGER.SYS,PUB” is the current session.

ftp> cd /SYS/WORK
250 CWD file action successful.

ftp> pwd
257-“/SYS/WORK” is the current directory.
257 “,MANAGER.SYS,PUB” is the current session.
```

Listing MPE/iX or POSIX Files

Earlier versions of the MPE/iX LISTF command may not report files in the POSIX name space; the LISTFILE command must be used instead. The FTP/iX client's LS and DIR commands provide a means to list the files on the remote host (refer to earlier sections of this manual for syntax and usage of these commands). If no file set is specified, the default is (./@). To list files located on the clients host, use either "`:listf <parms>`" or "`:listfile <parms>`". Refer to the *MPE/iX Commands Reference Manual* for command syntax and usage.

A POSIX file name can be fully qualified from the Root (/), qualified from the current working directory (./) or qualified from the parent of the current working directory (../).

Example 1:

This example specifies a file name qualified from the Root (/).

```
ftp> pwd
257-"/SYS/PUB" is the current directory.
257 ",MANAGER.SYS,PUB" is the current session.

ftp> dir /SYS/PUB/CATALOG
200 PORT command ok.
150 File: LISTFILE /SYS/PUB/CATALOG,2 opened; data connection will be opened

PATH= /SYS/PUB/
CODE ------------LOGICAL RECORD ------------      ---- SPACE ----   FILENAME

     SIZE    TYP    EOF    LIMIT    R/B      SECTORS   #X   MX
     80B      FA    13735  13735    16       4304      17    *      *CATALOG

226 Transfer complete.
228 bytes received in 0.01 seconds (24.74 Kbytes/sec)
```

Example 2:

This example specifies a file name qualified from the current working directory "cwd". It is the same file as used in the previous example.

```
ftp> pwd
257-"/SYS/PUB" is the current directory.
257 ",MANAGER.SYS,PUB" is the current session.

ftp> dir ./CATALOG
200 PORT command ok.
150 File: LISTFILE ./CATALOG,2 opened; data connection will be opened

PATH= /SYS/PUB/
CODE ------------LOGICAL RECORD ------------      ---- SPACE ----   FILENAME

     SIZE    TYP    EOF    LIMIT    R/B      SECTORS   #X   MX
     80B      FA    13735  13735    16       4304      17    *      *CATALOG

226 Transfer complete.
228 bytes received in 0.02 seconds (13.92 Kbytes/sec)
```

Bytestream Files

Bytestream record formats can exist in both the MPE/iX and POSIX namespace. The purpose of this new record format is to provide an emulated file format which exists on character-oriented file systems, where the user application is responsible for defining where logical records begin and end.

To create such a file, use the build parameters:

```
ftp> put <source-file> <target-file>;rec=,,b
```

The (b) references the bytestream record type. Note that no record size is given. This is because bytestream files always have 1 byte records; any other value would be overwritten. Similarly bytestream files are always MPE file type ASCII as opposed to Binary. Regardless of whether they are MPE file type ASCII or Binary, they may be transferred in either FTP ASCII or Binary transfer mode.

Build

Creates and immediately allocates a new empty file on disk.

The build command has been enhanced to include this bytestream record type.

Syntax

```
BUILD filereference

                              [F]
;REC[[recsize][,[blkfactor][,[U][,BINARY]]]]
                              [V][,ASCII ]
                              [B]
```

The “[B]” is the bytestream record type specification.

Example:

This sample of a bytestream file, a copy from CATALOG.PUB.SYS which is a (80) Byte per Record, Fixed ASCII file, to the bytestream record format which is a (1) Byte per Record Bytestream ASCII file.

```
:copy catalog.pub.sys,asciifix
:build bytestrm;rec=,,b;disc=2000000
:fcopy from=asciifix;to=bytestrm ;nouserlabels
HP31900A.05.02 FILE COPIER (c) HEWLETT-PACKARD CO. 1990

*200*WARNING: FROMFILE RECSIZE IS 80 BYTES, TOFILE RECSIZE IS 8192 BYTES.
CONTINUE OPERATION (Y OR N) ?yes
EOF FOUND IN FROMFILE AFTER RECORD 13819
13820 RECORDS PROCESSED *** 0 ERRORS

:listf ,2
ACCOUNT =    SYS    GROUP= WORK
FILENAME     CODE    --------LOGICAL RECORD--------      ----SPACE----

             SIZE    TYP   EOF      LIMIT      R/B         SECTORS  #X  MX
ASCIIFIX      80B     FA    13820    13820      16          4304     1   *
```

```
BYTESTRM       1B     BA     1119420  2000000     1          4608     18     *
```

NOTE The BYTESRM file contains 80 (record size) x 13820 (number of records) plus 13820 (the number of line feed (hexadecimal $0A) record terminator characters).

Building and Transferring a Bytestream File

These examples create a bytestream file in the POSIX shell and MPE namespace with the vi editor and then transfers that file to a remote system with FTP.

Example 1:

File transfer HP3000 MPE/iX to HP3000 MPE/iX system.

```
:sh
############################################################################

                  MPE/iX Shell and Utilities (A.00.00)

      COPYRIGHT (C) Hewlett-Packard Company 1992, All Rights Reserved.

############################################################################

shell/iX > vi
```

This is a test file with sample data.

```
-
:w bytestrm

shell/iX> ls bytestrm
bytestrm

shell/iX> /SYS/ARPA/FTP
File Transfer Protocol [A0006018] (C) Hewlett-Packard Co. 1990

ftp> :LISTFILE /SYS/PUB/bytestrm
PATH= /SYS/PUB/

bytestrm

ftp> pwd
257-"/SYS/WORK" is the current directory.
257 ",MANAGER.SYS,WORK" is the current session.

ftp> put /SYS/PUB/bytestrm
200 PORT command ok.
150 File: ./bytestrm;REC=,,B;DISC=2147483647,8 opened; data connection will
be opened
226 Transfer complete.
38 bytes sent in 0.00 seconds (18.55 Kbytes/sec)

ftp> dir ./bytestrm
```

```
200 PORT command ok.
150 File: LISTFILE .bytestrm,2 opened; data connection will be opened

PATH= /SYS/PUB/
     CODE    --------LOGICAL RECORD--------         ----SPACE----   FILENAME

     SIZE    TYP   EOF    LIMIT         R/B     SECTORS  #X  MX
      80B     FA    38    2147483647     1       16       1  32     bytestrm

226 Transfer complete.
230 bytes received in 0.01 seconds (24.96 Kbytes/sec)
```

Example 2:

File transfer HPUX (unix) system to HP3000 MPE/iX system.

```
vi bytestrm
```

This is a bytestrm test data file.

```
:wq
“bytestrm” [New file] 1 lines, 34 characters

ftp> put bytestrm ./bytestrm
200 PORT command ok.
150 File: ./bytestrm;rec=,,b opened; data connection will be opened
226 Transfer complete.
35 bytes sent in 0.00 seconds (64.73) Kbytes/sec)

ftp> dir ./bytestrm
200 PORT command ok.
150 File: LISTFILE ./bytestrm,2 opened; data connection will be opened

PATH= /SYS/PUB/
     CODE    --------LOGICAL RECORD--------         ----SPACE----   FILENAME

     SIZE    TYP   EOF    LIMIT         R/B     SECTORS  #X  MX
      1B      BA    35    2048007        1       16       1   8     bytestrm

226 Transfer complete.
```

NOTE When transferring a Bytestream file from UNIX to a MPE/iX system it is necessary to specify “;rec=,,b”. This is required as a result of FTP/iX supporting a default of “fixed” file type rather than bytestream which is the default on UNIX and POSIX systems.

Example 3:

File transfer HP 3000 MPE/iX system to HPUX (UNIX) system.

```
ftp> bytestream
200 Type set to L (byte size 8).
ftp> put ./bytestrm
200 PORT command successful.
150 Opening BINARY mode data connection for ./bytestrm.
226 Transfer complete.
35 bytes sent in 0.00 seconds (+INF Kbytes/sec)
```

```
ftp> dir ./bytestrm
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r--   1 jph     network     35 Jun 20 16:19 ./bytestrm
226 Transfer complete.
66 bytes received in 0.18 seconds (0.37 Kbytes/sec)
```

FTP Default Name Space

FTP default name space is “almost always” MPE name space when specifying files to be transferred with FTP commands (PUT or GET as examples). The only exception to this is the FTP MKDIR and RMDIR commands which only operate with POSIX directories; thus, if you did not specify the (./) POSIX prefix, FTP/iX specifies it for you.

For example, assume the POSIX current working directory “cwd” is: /SYS/WORK

FTP command	Result of FTP command
dir /SYS/PUB/CATALOG	Display existing file (MPE space)
put testdata	/SYS/WORK/TESTDATA (MPE space)
put testdata ./testdata	/SYS/WORK/testdata (posix space)
put testdata /SYS/WORK/testdata	/SYS/WORK/testdata (posix space)
put testdata longfilename	{failed – invalid MPE space name}
put testdata ./longfilename	/SYS/WORK/longfilename (posix)
put testdata name_name	{failed – invalid MPE space name}
put testdata ./name_name	/SYS/WORK/name_name (posix space)

Transferring Files

File names are taken as they are presented, as opposed to the directory commands described in previous sections (directory commands which preface (./) to the file name). Directory commands are always dealing with POSIX files, so prefacing a (./) is helpful to the user. When transferring files, files prefaced with a (.) or (/) will reference a file in the POSIX name-space as opposed to the MPE/iX name-space.

NOTE If the path specified for the source file does not exist on the target system, the target file must be specified in the FTP/iX file transfer command. If the target file is not specified, FTP/iX will attempt to create the file with the path specified. If the target file is to be copied to the current working directory, only the filename (not the full path) needs to be specified.

Example 1:

This example demonstrates a FTP data transfer from a file in a MPE group to a POSIX directory.

```
:listf testdata,2
ACCOUNT =    SYS    GROUP= WORK
FILENAME     CODE    --------LOGICAL RECORD--------         ----SPACE----

             SIZE    TYP   EOF      LIMIT      R/B          SECTORS  #X  MX
TESTDATA     80B     FA    13820    13820      16           4304     1   *
```

```
ftp> pwd
257-"/SYS/WORK" is the current directory.
257 ",MANAGER.SYS,WORK" is the current session.
```

```
ftp> mkdir temp
257 MKD successful, "temp" created.
```

```
ftp> cd temp
250 CWD file action successful.
```

```
ftp> pwd
257-"/SYS/WORK/temp" is the current directory.
257 ",MANAGER.SYS,WORK" is the current session.
```

```
ftp> put testdata
200 PORT command ok.
150 File: testdata;REC=-80,16,F,ASCII;DISC=13820,8 opened; data connection
will be opened
226 Transfer complete.
1133496 bytes sent in 1.25 seconds (885.54 Kbytes/sec)
```

```
ftp> dir
200 PORT command ok.
150 File: LISTFILE ./@,2 opened; data connection will be opened

     PATH= /SYS/PUB/
     CODE    --------LOGICAL RECORD--------         ----SPACE----   FILENAME

     SIZE    TYP   EOF    LIMIT         R/B     SECTORS  #X  MX
     80B      BA   13820  13820         16       4320    17   8     TESTDATA

226 Transfer complete.
235 bytes received in 0.02 seconds (14.34 Kbytes/sec)
```

Example 2:

This example demonstrates a FTP data transfer from a file in a POSIX directory to a POSIX directory. The sample file copied in the prior example will be copied back to the local system in the POSIX directory /SYS/WORK/temp2.

```
ftp> :newdir /SYS/WORK/temp2
Created directory "/SYS/WORK/temp2"
```

```
ftp> :chdir /SYS/WORK/temp2
CWD is "/SYS/WORK/temp2".
```

```
ftp> :showvar HPCWD
HPCWD = /SYS/WORK/temp2

ftp> :SHOWME
USER: #S6,MANAGER.SYS,PUB           (IN PROGRAM)

ftp> PWD
257-”/SYS/WORK/temp” is the current directory.
257 “,MANAGER.SYS,WORK” is the current session.

ftp> DIR
200 PORT command ok.
150 File: LISTFILE ./@,2 opened; data connection will be opened

     PATH= /SYS/WORK/temp/
     CODE    --------LOGICAL RECORD--------         ----SPACE----   FILENAME

     SIZE    TYP   EOF    LIMIT         R/B     SECTORS  #X  MX
     80B      FA   13820  13820         16       4320    17   8     TESTDATA

226 Transfer complete.
235 bytes received in 0.02 seconds (12.75 Kbytes/sec)

ftp> get testdata
200 PORT command ok.
150 File: testdata opened; data connection will be opened
226 Transfer complete.
1133496 bytes received in 2.19 seconds (504.30 Kbytes/sec)
ftp> :listfile,2

     PATH= /SYS/WORK/temp2/
     CODE    --------LOGICAL RECORD--------         ----SPACE----   FILENAME

     SIZE    TYP   EOF    LIMIT         R/B     SECTORS  #X  MX
     80B      FA   13820  13820         16       4320    17   8     TESTDATA
```

Example 3:

This example demonstrates a FTP data transfer utilizing POSIX name space.

```
:listf testdata,2
ACCOUNT =    SYS    GROUP= WORK
FILENAME     CODE    --------LOGICAL RECORD--------         ----SPACE----

             SIZE    TYP   EOF      LIMIT      R/B         SECTORS  #X  MX
TESTDATA     80B     FA    13820    13820      16          4304     1   *

ftp> pwd
257-“/SYS/WORK” is the current directory.
257 “,MANAGER.SYS,WORK” is the current session.

ftp> put TESTDATA /SYS/WORK/temp/long_file_name
200 PORT command ok.
150 File: /SYS/WORK/temp/long_file_name;REC=-80,16,F,ASCII;DISC=13820,8
opened:
data connection will be opened
226 Transfer complete.
1133496 bytes sent in 1.29 seconds (858.09 Kbytes/sec)
```

When specifying files in the POSIX file space, you must specify the file in POSIX syntax or else FTP will attempt to fit the file in the MPE file space and fail, as in the following examples:

Example 4:

```
ftp> put TESTDATA long_file_name2
```

```
200 PORT command ok.
550 An invalid character was found in an MPE syntax name. (FILE SYSTEM ERROR
-102)
Data Transfer Request Failed. (FTPERR 13)
```

Example 5:

```
ftp> put TESTDATA longfilename12
```

```
200 PORT command ok.
500 The file name specified exceeds the maximum legal length. (FILE SYSTEM
ERROR -110)
Data Transfer Request Failed. (FTPERR 13)
```

B Error Messages

This appendix contains messages and errors that may be received if you are using FTP to access another host or an FTP server.

FTP User Messages and Errors

These error and warning messages may be returned to FTP users on MPE/iX while using the FTP user program.

In the following list of messages and errors, the exclamation point (!) represents a variable value.

(FTPERR 3) **MESSAGE: Error in loading keywords from FTP catalog.**

CAUSE: The User-FTP program encountered an error while trying to load the FTP command keywords onto its program stack from the FTP catalog. This probably indicates an internal FTP error.

ACTION: Check the permissions on the file FTPC000.ARPA.SYS. The security for READ must be set to ANY. If the permissions are correct, note the circumstances and contact your Hewlett-Packard representative for assistance.

(FTPERR 5) **MESSAGE: Command must be less than or equal to 255 characters.**

CAUSE: The user entered an FTP command which exceeded 255 characters.

ACTION: Try entering the command using fewer characters.

(FTPERR 6) **MESSAGE: ?Invalid command**

CAUSE: An invalid or unknown command was issued to the User-FTP program.

ACTION: Check the command for spelling. Use the HELP command for a list of supported HP ARPA FTP commands.

(FTPERR 8) **MESSAGE: Not connected.**

CAUSE: An FTP command was entered which requires a host connection and no connection was established.

ACTION: Open a host connection using the OPEN command and try the command again.

(FTPERR 9) **MESSAGE: Cannot connect to host: !**

CAUSE: FTP was unable to open a host connection. Typically this FTP error is accompanied by a socket error which provides more information.

ACTION: Check the spelling of the host name or IP address. Otherwise, seek the assistance of your Node Manager.

(FTPERR 10) **MESSAGE: Invalid Command Syntax.**

CAUSE: An FTP command was entered with invalid parameter syntax.

ACTION: Check the syntax of the desired command in the *HP ARPA FTP User's Guide*.

(FTPERR 11) **MESSAGE: Invalid reply code from remote server.**

CAUSE: An erroneous reply code was returned from the remote server for some FTP command.

ACTION: Enable debug and verbose and try the operation again.

(FTPERR 13) **MESSAGE: Transfer Request Failed.**

CAUSE: An error was detected during a data transfer. This error is accompanied by another error which indicates the problem.

ACTION: Use the accompanying error to resolve the problem.

(FTPERR 14) **MESSAGE: Target File Could Not Be Opened.**

CAUSE: The MPE/iX target file could not be opened. This error is typically accompanied by a file system error.

ACTION: Use the accompanying file system error to resolve the problem.

(FTPWARN 15) **MESSAGE: Some records were truncated during transfer.**

CAUSE: A data transfer to MPE/iX resulted in some records being truncated in the target file.

ACTION: It may be necessary to increase the maximum record size using build parameters.

(FTPERR 16) **MESSAGE: End of File encountered on target file during transfer. Target file not saved. Increase file limit and try again.**

CAUSE: A data transfer to an MPE/iX target file did not have enough records to hold the data from the source file.

ACTION: It may be necessary to increase the maximum number of records in the target file by using build parameters.

(FTPERR 17) **MESSAGE: End of File encountered on target file during transfer. Data not & appended to target file. Copy target file to new file with a larger & limit and append to the new file.**

CAUSE: A data transfer (in APPEnd mode) to an MPE/iX target file, did not have enough records to hold the extra data from the source file.

ACTION: Copy the target file to a new file with a larger limit and append to the new file.

(FTPWARN 18) **MESSAGE: We only support stream mode, sorry.**

CAUSE: The FTP MODE command was entered.

ACTION: None.

(FTPWARN 19) **MESSAGE: We only support file structure, sorry.**

CAUSE: The FTP STRUCT command was entered.

ACTION: None.

(FTPWARN 20) **MESSAGE: We only support non-print format, sorry.**

CAUSE: The FTP FORM command was entered.

ACTION: None.

(FTPINFO 21) **MESSAGE: Verbose mode off.**

CAUSE: The FTP VERBOSE command was entered when verbose was enabled.

ACTION: None.

(FTPINFO 22) **MESSAGE: Verbose mode on.**

CAUSE: The FTP VERBOSE command was entered when verbose was disabled.

ACTION: None.

(FTPINFO 23) **MESSAGE: Debugging off (debug=0).**

CAUSE: The FTP DEBUG command was entered when debug was enabled.

ACTION: None.

(FTPINFO 24) **MESSAGE: Debugging off (debug=1).**

CAUSE: The FTP DEBUG command was entered when debug was disabled.

ACTION: None.

(FTPINFO 25) **MESSAGE: Using ascii mode to transfer files.**

CAUSE: The FTP TYPE command was entered without any parameters while in ASCII mode.

ACTION: None.

(FTPINFO 26) **MESSAGE: Using binary mode to transfer files.**

CAUSE: The FTP TYPE command was entered without any parameters while in binary mode.

ACTION: None.

(FTPWARN 27) **MESSAGE: Already connected to !; use user or close first.**

CAUSE: The user entered the FTP OPEN command during an opened session.

ACTION: Enter the USER command to logon, or the CLOSE command before opening another connection.

(FTPERR 28) **MESSAGE: Unknown Error Detected. Subsystem = !, Information = !.**

CAUSE: An unknown error was detected from some underlying subsystem.

ACTION: Seek the assistance of your Hewlett-Packard representative.

(FTPERR 29) **MESSAGE: We only support MPEXL Fixed and Variable record format, sorry.**

CAUSE: The user specified a BUILD command parameter for record type that was not Fixed or Variable.

ACTION: Only fixed and variable record format is supported.

(FTPERR 30) **MESSAGE: We only support MPEXL Standard files, sorry.**

CAUSE: The user specified a build parameter for a file type that is not of type Standard (STD).

ACTION: Only the STD file type is supported.

(FTPERR 31) **MESSAGE: We don’t allow MPEXL file equates, sorry.**

CAUSE: The user attempted to use a file equate during a file transfer.

ACTION: File equates are not yet supported. Try the transfer again without using file equates.

(FTPERR 32) **MESSAGE: Remote server received timeout. Closing connection.**

CAUSE: The FTP user did not receive a required reply from an FTP server in the allocated time window.

ACTION: Close the connection and quit before trying again. Try to ping the remote system using PING.NET.SYS. If this does not succeed, no FTP connection can be established. Follow the troubleshooting instructions in the *Installing and Managing HP ARPA File Transfer Protocol Network Manager's Guide*.

(FTPINFO 33) **MESSAGE: ExitOnError mode on.**

CAUSE: The EXITONERROR command was entered while the exit-on-error feature was disabled.

ACTION: None.

(FTPINFO 34) **MESSAGE: ExitOnError mode off.**

CAUSE: The EXITONERROR command was entered while the exit-on-error feature was enabled.

ACTION: None.

(FTPERR 35) **MESSAGE: Invalid IP address.**

CAUSE: An invalid IP address was used while trying to open a connection.

ACTION: Refer to the body of this manual for assistance.

(FTPWARN 36) **MESSAGE: Performing an ASCII transfer to a binary file.**

CAUSE: An ASCII data transfer occurred to an MPE/iX binary file.

ACTION: If the intent was to transfer binary data, then use the BINARY command to change transfer mode to binary. If the intent was to transfer ASCII data, then specify an MPE/iX ASCII file using the appropriate MPE/iX BUILD parameter.

(FTPWARN 37) **MESSAGE: Performing a binary transfer to a binary file.**

CAUSE: A Binary data transfer occurred to an MPE/iX ASCII file.

ACTION: If intent was to transfer ASCII data, then use the ASCII command to change transfer mode to ASCII. If the intent was to transfer binary data, then specify an MPE/iX binary file using the appropriate MPE/iX BUILD parameter. Refer to the *HP ARPA File Transfer Protocol User's Guide* for further assistance.

(FTPERR 38) **MESSAGE: Error opening data connection.**

CAUSE: Internal Error. An I/O completion was received for an unknown file number during the attempted establishment of a data connection.

ACTION: Note the circumstances and seek the assistance of your Hewlett-Packard representative.

(FTPERR 39) **MESSAGE: ?Ambiguous command.**

CAUSE: A non-unique command string was entered at the ftp> prompt.

ACTION: Check the command spelling and try again.

(FTPINFO 40) **MESSAGE: Connected to !.**

CAUSE: A connection request to a remote FTP server was accepted.

ACTION: None.

(FTPINFO 41) **MESSAGE: Interactive mode on.**

CAUSE: The PROMPT command was issued while prompting was disabled.

ACTION: None.

(FTPINFO 42) **MESSAGE: Interactive mode off.**

CAUSE: The PROMPT command was issued while prompting was enabled.

ACTION: None.

(FTPINFO 43) **MESSAGE: Lower case on.**

CAUSE: The CASE command was issued while “Forcing Lower Case” was disabled.

ACTION: None.

(FTPINFO 44) **MESSAGE: Lower case off.**

CAUSE: The CASE command was issued while “Forcing Lower Case” was enabled.

ACTION: None.

(FTPERR 45) **MESSAGE: File accessed exclusively.**

CAUSE: The file is already being accessed exclusively by the system or by another user.

ACTION: If the file is being accessed by the system, there is nothing you can do. If the file is being accessed by another user, try again later.

(FTPERR 46) **MESSAGE: File name too long.**

CAUSE: The filename specified is longer than the maximum allowed by MPE: “FILENAME/LOCKWORD.GROUPNAM.ACCTNAME”. As of release 4.5, MPE/iX supports longer filenames, but unless the first character of that name is a dot (.) or a slash (/), the name will be interpreted as an MPE name.

ACTION: Specify a file name that is shorter or starts with a dot or a slash.

(FTPERR 47) **MESSAGE: Indirect file not found.**

CAUSE: The indirect file name specified in the mput command was not found: “mput ^filename”.

ACTION: Make sure the indirect file exists.

(FTPERR 48) **MESSAGE: Could not change directory to “!”.**

CAUSE: The directory specified does not exist or is not accessible.

ACTION: Check to see if the directory exists and if so, what its access rights are.

(FTPINFO 49) **MESSAGE: Connection time-out value set to ! seconds.**

CAUSE: User entered "TIMEOUT [num-secs]"

ACTION: If num-secs was specified, the connection time-out value was set. Otherwise, the current value was simply displayed.

Note: a value of 0 indicates no time-outs.

(FTPINFO 50) **MESSAGE: Store unique mode on.**

CAUSE: The FTP sunique command was entered when store-unique was disabled.

ACTION: Subsequent put commands will guarantee that the file will be created with a unique name on the remote machine.

(FTPINFO 51) **MESSAGE: Store unique mode off.**

CAUSE: The FTP sunique command was entered when store-unique was enabled.

ACTION: None.

(FTPINFO 52) **MESSAGE: Receive unique mode on.**

CAUSE: The FTP runique command was entered when receive-unique was disabled.

ACTION: Subsequent get commands will guarantee that the file will be created with a unique name on the local machine.

(FTPINFO 53) **MESSAGE: Receive unique mode off.**

CAUSE: The FTP runique command was entered when receive-unique was enabled.

ACTION: None.

(FTPINFO 54) **MESSAGE: Command tracing on.**

CAUSE: The FTP trace command was entered when command tracing was disabled.

ACTION: When an FTP connection to a remote system is established, TCP tracing will be turned on. If a connection already exists, tracing will be turned on now.

(FTPINFO 55) **MESSAGE: Command tracing off.**

CAUSE: The FTP trace command was entered when command tracing was enabled.

ACTION: If an FTP connection exists to a remote system and tracing is enabled, it will be turned off.

(FTPINFO 56) **MESSAGE: Data tracing on.**

CAUSE: The FTP trace command was entered when data tracing was disabled.

ACTION: When a connection for a data transfer is established, TCP tracing will be turned on.

(FTPINFO 57) **MESSAGE: Data tracing off.**

CAUSE: The FTP trace command was entered when data tracing was enabled.

ACTION: None.

(FTPINFO 58) **MESSAGE: Command tracing file name is !.**

CAUSE: Command tracing was turned on and a connection to a remote system was established.

ACTION: Use this file for debugging purposes.

(FTPINFO 59) **MESSAGE: Data tracing file name is !.**

CAUSE: Data tracing was turned on and a data transfer connection was established.

ACTION: Use this file for debugging purposes.

(FTPINFO 60) **MESSAGE: Using byte-stream mode to transfer files.**

CAUSE: The FTP type command was entered without any parameters while in byte-stream mode.

ACTION: None.

(FTPINFO 61) **MESSAGE: Unique file name cannot be created.**

CAUSE: Either the file name was so long that appending (.1) surpassed the maximum legal file name length, or every appendage from (.1 up to .99) was tried and no unique name could be found.

ACTION: Try a different, possible shorter, file name.

(FTPERR 63) **MESSAGE: Append mode not valid with record or file type.**

CAUSE: Attempt to transfer a non-standard MPE file in append mode.

ACTION: Do not transfer non-standard MPE files in append mode.

FTP Server Messages and Errors

FTP users accessing an MPE/iX-based FTP server may receive the messages and/or errors listed below which are based on the *Military Standard*, FTP, MIL-STD-1780.

The type of reply is based on the first digit of the number:

Number	Type
1nn	Positive preliminary reply
2nn	Positive completion reply
3nn	Positive intermediate reply
4nn	Transient negative reply which user may wish to retry
5nn	Permanent negative reply

The second digit in the message number represents the following function groupings (where the first digit x is from 1 to 5 as described above):

Number	Type
x0n	Syntax errors
x1n	Replies to requests for information (status, help)
x2n	Replies referring to the control and data connections
x3n	Replies for logon process and accounting procedures
x4n	Unspecified
x5n	Status of the server file system

In the following list of messages and errors, the exclamation point (!) represents a variable value.

125 MESSAGE: Data connection already open; transfer starting

CAUSE: Data connection has been opened, data transfer has started.

ACTION: None.

150 MESSAGE: File ! opened; data connection will be opened

CAUSE: The file requested for a GET, LS or DIR has been opened.

ACTION: None.

200 MESSAGE: ! command ok.

CAUSE: The command was accepted by the server.

ACTION: None.

200 MESSAGE: Type set to !.

CAUSE: A TYPE command was received and accepted.

ACTION: None.

202 **MESSAGE: Command not implemented at this site.**

CAUSE: The MPE/iX FTP server received a command that is not implemented.

ACTION: This command cannot be used.

213 **MESSAGE: "!": !.**

CAUSE: A file status (SIZE | MDTM) command was received and accepted. This first parameter is the filename. The second contains detailed information that differs depending on the command.

ACTION: None.

215 **MESSAGE: MPE/iX system type.**

CAUSE: The server received a SYST command.

ACTION: None.

220 **MESSAGE: HP ARPA File Transfer Protocol Server [!] (C) Hewlett-Packard Co. 1990**

CAUSE: The server has been started and is ready for user commands.

ACTION: Enter FTP commands.

221 **MESSAGE: Server is closing command connection**

CAUSE: The server received a QUIT command and is shutting down.

ACTION: None.

226 **MESSAGE: Transfer complete.**

CAUSE: A file transfer for a GET, PUT, LS or DIR has completed.

ACTION: None.

225 **MESSAGE: Data connection is opened with no transfer in progress.**

CAUSE: NOT USED

ACTION: None.

226 **MESSAGE: Transfer complete. Some records were truncated during transfer.**

CAUSE: A data transfer to MPE/iX resulted in some records being truncated in the target file.

ACTION: It may be necessary to increase the maximum record size using build parameters. Review the *HP ARPA FTP User's Guide* for more information on this subject.

230 **MESSAGE: User logged on**

CAUSE: A USER command was received and accepted. The logon has succeeded.

ACTION: None.

250 **MESSAGE: ! File action successful.**

CAUSE: A file action for DELE/RMD/XRMD/CWD/XCWD has competed.

ACTION: None.

257 **MESSAGE: ! is the current directory. "!" is the current session.**

CAUSE: A PWD command was received and accepted.

ACTION: None.

257 **MESSAGE: ! successful, “!” created**

CAUSE: A directory creation (MKD/XMKD) has completed.

ACTION: None.

331 **MESSAGE: Password required for !. [userpass][,acctpass][,grouppass]**

CAUSE: A USER command has been received, the server is waiting for the password.

ACTION: Enter the passwords required.

350 **MESSAGE: File exists, ready for destination name.**

CAUSE: An RNFR (rename from) command has been received, the server is waiting for a RNTO (rename to). This is part of the user RENAME command.

ACTION: None.

426 **MESSAGE: Data transfer aborted**

CAUSE: The data connection could not be opened or there was a network error during the file transfer.

ACTION: Try the transfer again. If this doesn’t work try exiting FTP and connecting again, then transferring the file. If that doesn’t work check for a networking problem (resources, connection establishment, etc.) on both systems.

500 **MESSAGE: ‘!’: Unknown command**

CAUSE: The MPE/iX FTP server received an unrecognized command.

ACTION: This command cannot be used.

501 **MESSAGE: Syntax error in parameter !**

CAUSE: One of the parameters in the command was not correct. The ! will contain the parameter in question.

ACTION: Check the parameter and try again.

502 **MESSAGE: ! Command not implemented by this server.**

CAUSE: The MPE/iX FTP server received a command that it recognized but has not implemented.

ACTION: This command cannot be used.

503 **MESSAGE: Illegal sequence of commands**

CAUSE: The server was sent a series of commands that were in the wrong order (example: RNTO before RNFR).

ACTION: Check the commands and send them in the correct order.

504 **MESSAGE: Command not implemented for parameter !**

CAUSE: A parameter was specified that is not supported. For example TYPE EBCDIC is NOT supported by the MPE/iX FTP server.

ACTION: Specify a parameter which is supported.

530 MESSAGE: User log on unsuccessful

CAUSE: The server received a user account and/or password which were not valid.

ACTION: Specify the correct user account and password. If the user is sure that they are correct, notify the network manager and have him/her refer to the *Installing and Managing HP ARPA File Transfer Protocol Network Manager's Guide*.

530 MESSAGE: Please login with USER and PASS.

CAUSE: The user entered a command without being logged onto the FTP server.

ACTION: Log on with the USER command, then enter the command again.

550 MESSAGE: File request not taken: !

CAUSE: An MPE/iX File System error was received. The parameter will contain the text of the specific File System error.

ACTION: Refer to the *MPE/iX Error Messages Manual (Volume 1, 2 and 3)*.

553 MESSAGE: Illegal file name

CAUSE: The file name specified was not a valid MPE/iX file name.

ACTION: Use the correct file name format: either `filename` or `filename.group.account` where each file name part is up to eight alphanumeric characters, beginning with an alphabetic character.

Glossary

A

address A numerical identifier defined and used by a particular protocol and associated software to distinguish one node from another.

address resolution In NS networks, the mapping of node names to IP addresses and the mapping of IP addresses to subnet addresses. *See Also:* **probe protocol**, **ARP**.

ARP Address Resolution Protocol. ARP provides IP to LAN station address resolution for Ethernet nodes on a LAN.

ARPA Advanced Research Projects Agency.

ASCII American National Standard Code for Information Interchange. A character set using 7-bit code used for information interchange among data processing and data communications systems. The American implementation of International Alphabet No. 5.

B

binary mode Data transfer scheme in which no special character processing is performed. All characters are considered to be data and are passed through with no control actions being taken.

byte A sequence of eight consecutive bits operated on as a unit.

bytestream A file record format in which the data definition is defined by the application. Data is stored in a contiguous stream of bytes (no record terminators).

C

control connection The communication path between the FTP User PI and the Server PI used to exchange commands and replies. This connection follows the TELNET Protocol guidelines.

D

data connection A full duplex connection over which data is transferred in a specified mode and type. The path may be between a serve-DTP and a user-DTP, or between two server DTPs.

directory See working directory.

DTP The data transfer process that establishes and manages the data connection.

E

environment A session that is established on a remote node.

Ethernet Local Area Network system that uses baseband transmission at 10 Mbps over coaxial cable. Ethernet is a trademark of Xerox Corporation.

F

file equation Assignment statement used to associate a file with a specific device or type of device during execution of a program.

file system The part of the operating system that handles access to input/output devices, data blocking, buffering, data transfers, and deblocking.

flow control A means of regulating the rate at which data transfer takes place between devices to protect against data overruns.

FTP File Transfer Protocol. The application protocol offering file service in the Internet suite of protocols developed by the Advanced Research Projects Agency (ARPA).

H

host computer The primary or controlling computer on a network. The computer on which the network control software resides. For HP purposes, it may also be used to distinguish the MPE/iX system (host) from the DTC.

I

IEEE 802.3 A standard for a broadcast local area network published by the Institute for Electrical and Electronics Engineers (IEEE). This standard is used for both the ThinLAN and ThickLAN implementations of the LAN.

Internet Protocol (IP) A protocol used to provide routing between different local networks in an internetwork, as well as among nodes in the same local network. *See also* **IP address**.

intrinsic System call or procedure accessible by user programs which provides an interface to operating system resources and functions. Intrinsics perform common tasks such as file access and device control.

IP Internet Protocol. A protocol used to provide routing between different local networks in an internetwork, as well as among nodes in the same local network. The internet protocol corresponds to layer 3, the network layer, of the OSI model. *See also* **IP address**.

IP address Internet Protocol address. An address used by the Internet Protocol to perform internet routing. A complete IP address comprises a network portion and a node portion. The network portion of the IP address identifies a network, and the node portion identifies a node within the network.

L

loopback The routing of messages from a node back to itself.

M

MPE/iX The operating system for the HP 3000 Series 900 computers.

N

network address This can be either 1) the network portion of an IP address as opposed to the node portion, or 2) when referring to X.25 networks, it is a node's X.25 address.

network directory A file containing information required for one node to communicate with other nodes in 1) an internetwork, 2) an X.25 network, or 3) a network that contains non-HP nodes. The active network directory on a node must be named `NSDIR.NET.SYS`.

NMMGR Node Management Services Configuration Manager. A software subsystem that enables you to configure DTC connectivity and network access parameters for an HP 3000 Series 900 computer.

NMMGRVER Node management services conversion utility. A conversion program that converts configuration files created with NMMGR from an earlier version to the latest format.

node A computer that is part of a network. The DTC is also considered to be a node and has its own address.

node address The node portion of an IP address.

Node Management Services Configuration Manager *See* **NMMGR**.

node name A character string that uniquely identifies each system in a network or internetwork. Each node name in a network or internetwork must be unique; however, a single node can be identified by more than one node name.

NS 3000/iX Link Software and hardware that provides the connection between nodes on a network. Some of the NS 3000/iX links available are the ThinLAN 3000/iX Link and its ThickLAN option, the DTC/X.25 iX Network Link, and the NS Point-to-Point 3000/iX Link.

NS 3000/iX Network Services Software applications that can be used to access data, initiate processes, and exchange information among nodes in a network. The services are RPM, VT, RFA, RDBA, and NFT.

NSDIR.NET.SYS Name of the active network directory file. *See also* **network directory**.

P

privileged mode A capability assigned to accounts, groups, or users allowing unrestricted memory access, access to privileged CPU instructions, and the ability to call privileged procedures.

probe protocol An HP protocol used by NS 3000/iX IEEE 802.3 networks to obtain information about other nodes on the network. It provides name to IP address resolution, and IP to IEEE 802.3 address resolution.

protocol A set of rules that enables two or more data processing entities to exchange information. In networks, protocols are the rules and conventions that govern each layer of network architecture. They define what functions are to be performed and how messages are to be exchanged.

PI Protocol Interpreter. In the FTP client-server model, describes the processing of FTP commands and replies over the control connection between the user and server.

R

remote node A node on an internetwork other than the node you are currently using or referring to.

V

Virtual Terminal A network service that allows a user to establish interactive sessions on a node.

W

working directory While using FTP, the directory on the remote system currently being accessed. Terminology used on UNIX and MS DOS systems. On MPE/iX, a comparable structure is group.

Index