

*** * * DRAFT * * ***

DDN FTP/HP3000

External Specifications



Information Networks Division

Location Code: 66-7860

Project Number: 6651-1131

February 10, 1986

David St. John

*** HP Confidential ***

PRODUCT IDENTIFICATION

SECTION

1

Name	HP/3000 DDN Services: FTP
Mnemonic	none
Project Number	6651-1131
Project Manager	Doug Heath
Project Engineer	David St. John
Product Manager	Dennis King
Product Assurance	Karen Dillon
Documentation	Mike Genevro
Off-Line Support	Bruna Byrne
On-Line Support	Susan Gennrich Lorraine Mehtens Jim Zepp

2.1 PRODUCT OVERVIEW

File Transfer Protocol will allow users to manipulate files on any remote host which also provides the service. Those manipulations include the storing or retrieving of files, the deletion of remote files, and the renaming of files on remote hosts. Many vendors have already provided this service to their customers. Therefore FTP allows HP customers the ability to transfer files not only between various types of HP systems that provide the service, but also between non-HP machines.

2.2 USER DEFINITIONS

Two users are defined: 1) the local user and 2) the remote user. The FTP service must have been previously allowed via the NSCONTROL command on both HP3000 systems, or some similar mechanism on other systems.

2.2.1 Local User

On the local system the FTP user must have established a session or job. The user can then access the FTP program by either running the FTP program or by using the CREATEPROCESS intrinsic for programmatic access. It is the responsibility of the user to specify a well-known name for the remote host to be accessed. That DDN host name must be in the Network Directory or obtainable by a Probe query. The user must then pass information to the remote system containing accounting information necessary to create a session on that system. No other actions are possible with the FTP program until the remote session has been created, except to close the connection or to use the HELP facility. All MPE and file system security restrictions are in effect for the local user.

2.2.2 Remote User

A session must be established on the HP3000 before a foreign user can transfer or access any files. The user will be subject to the usual MPE and file system securities as any local user on the same system. The remote session will remain until either the foreign user explicitly closes the connection or exits the local FTP program.

2.3 PRODUCT ENVIRONMENT

FTP will be operative in a normal HP3000 environment. There must be a physical connection to the DDN network through an INP or to a LAN through a LANIC board.

The software dependencies are several for FTP.

- 1) The DDN architectural model has X.25 at the network level. The X.25 must be certified with the

Product Specifications

DDN as compatible. X.25 is not necessary for use of the product on a LAN.

- 2) Level 4 specifications require TCP and IP. These must be present on the system for FTP to operate.
- 3) Altered NS session services are necessary. FTP access for users must be allowed via the NSCONTROL command and conversely can be disallowed. DSDAD and other utilities must also be accessible.
- 4) Network Directory and Network Manager should be present on the 3000 for resolving node name to addresses, thus providing connection to the remote host through the network.

All of these software modules must be present for FTP to operate. This does not include the dependencies which any of the above may have. DDN Services/HP3000: Investigation Report should be read for more detail.

2.4 USER DOCUMENTATION REQUIREMENTS

Since this service is a part of DDN services (see DDN Services/HP3000: Investigation Report) it should be included in a new Users/Programmers Manual for DDN Services. No updates are required to any MPE manual. Communications Handbook should have a section added for DDN Services, including FTP. Updates to other manuals required by changes in our dependencies are not covered in this document.

FTP should be included in any training planned by Off-line Support for DDN, whether it be SE training or that aimed at the end users.

FUNCTIONAL SPECIFICATIONS

SECTION

3

The stated objective of File Transfer Protocol, as outlined in Military Standard: File Transfer Protocol, is "1) to promote sharing of files..., 2) to encourage indirect or implicit...use of remote computers, 3) to shield a user from variations in file storage systems among hosts, and 4) to transfer data reliably and efficiently." This section details how FTP service on the HP3000 will meet these objectives.

3.1 USER INTERFACE

The commands that are discussed are strictly user commands. Although they may overlap, they should not be confused with FTP commands as outlined in the Mil_Std. Also the use of "user" and "server" in this document refer to the FTP modules. The user process will be the local process created on the local machine; the server will be a process to handle FTP requests from remote users.

3.1.1 Interactive Interface

FTP will be a program file which the user can activate by using the CI command:

```
RUN FTP.NET.SYS[INFO="remote host name"]
```

The optional INFO string will be used to establish a connection to the specified remote host; otherwise the user must specify the remote host name as the first executable command in the FTP subsystem other than HELP. This will allow the user to have a User Defined Command "FTP *remote host name*" which will give an appearance similar to many implementations. It was decided to not make "FTP" a MPE command, similar to "DSCOPY," so that the release of this product was not dependent upon changes to the operating system.

The user has several commands at his or her disposal. The command string can be up to 255 characters. The "&" character should be used before a carriage return to indicate that the command will be continued onto the next line. The syntax conventions used in this part are:

KEYWORDS of the command set are in bold letters (e.g., **GET**).

Arguments to the commands are in italics and are to be supplied by the user (e.g., **GET** *remote file name*).

Open and close brackets are used for optional fields (e.g., **GET** *remote file name* [*local file name*]).

3.1.1.1 User Command Set

OPEN *host*

This command must be the first command after entering the FTP program if the remote host name was not specified in an INFO string. All other commands before this one will return an error informing the user that the connection must be made first via the OPEN command. The exception to this is the HELP command below. The DDN host name must be an entry in the Network Directory or obtainable via a Probe query. If not the user must contact the Network Manager to enter it into the directory. For first release, only one connection can be open at one time.

Functional Specifications

USER *logon*

This command must be the first entered after the connection is made to the remote host, except the HELP command. The logon string must contain any information that is necessary to create a session on the remote node. The information will not be checked by the local FTP subsystem since the remote system type may be unknown to the FTP user program. If the logon string is incorrect and rejected by the remote FTP server, then the user will be notified and may re-enter the command. If the remote FTP server responds with a code 331, the user will be prompted for password(s). This remote FTP server will differentiate between user, group, and account passwords within the text to its 331 reply. If the server replies with a code 332, the user will be prompted for an account name; this implementation will not use a reply code 332. The user may enter a new USER command at any time which has the effect of logging off and logging on to the same remote node. The connection will not be closed. It is the user's responsibility to have one or more accounts on the remote host and to know what information must be passed to it.

CLOSE

The connection to a remote host will be closed and the remote session logged off, but the FTP user subsystem will not terminate. The user must use the OPEN command again to the same or a different host before continuing.

EXIT

The FTP user subsystem will be exited and the user returned to the CI level. If any connection is open, it will be closed before the program terminates.

HELP

The help facility will display the commands available to the user of the FTP subsystem, the syntax to be used, and the effect of each one. The help facility can be used whenever there is a subsystem prompt whether a connection is open or not.

TYPE {ASCII | BINARY}

This command specifies the type of data which will be transferred between systems. ASCII is the default, but the user will be able to switch the type via this command. ASCII and binary are the only two options available on first release. The general file system interface is addressed in a later section of this report.

GET *remote file name* [*local file name*]

This command transfers a file from a remote host to the local host. The remote file name must be specified; if the optional local file name is not present then the same remote file name will be used. Note that all file names on the HP3000 must meet the usual criteria.

PUT *local file name* [*remote file name*]

This is the mirror command of GET. It transfers a file from the local HP3000 to the remote system. The remote file name is optional and will default to the local file name if not included. Other restrictions outlined under the GET command also apply to this command.

APP[END] *source file*[:*host*] [*target file*]

The APPEND command will append either a remote file to a local file or a local file to a remote file. The source file name must always be included. If the host name option does not follow, it will be assumed that the specified file is a local file. If the remote host name is included, it must be as specified in either the OPEN command or in the INFO string of the MPE RUN command. If the target file name is not given, the assumption will be made that it is the same as the source file name. The target will be assumed to be on the remote host if the source file host option was not included. As stated in the Mil-Std, the file will be created if it does not exist; otherwise the source file will be appended to the target file.

RENAME *old file name , new file name*

Renaming of a file on a remote host can be executed using the familiar MPE command.

PURGE *file name*

The file will be deleted from the remote file system.

LISTF [*filename*]

A listing of remote files will be returned to the user. If no file name is specified, a list of files of the working directory on the remote host will be returned. If an argument is specified, information will be returned about that file or group of files. The file name argument will not be checked by the local 3000. This means that the user is responsible for knowing any meta-characters used by the remote file system. The remote HP3000 FTP server will pass the parameter directly to MPE for processing. This means that fileset options 0,1,2,-1 can be used as well.

3.1.1.2 Break and Control-Y

As a program file there should be no problems allowing the user to break the program and return to the CI. The FTP user process will be inactive until the RESUME command returns control to it. The control-y subsystem break will be used to either abort the data transfer or to return data transfer status to the user while it is in progress. The user will be able to decide which of these options are desired. The status option can be used to inform the user if the remote is not responding. Since the LISTF command is actually a data transfer command, the user can abort the remote file listing by using the control-y mechanism.

3.1.2 Programmatic Interface

Programmatic access to FTP will be via the MPE CREATEPROCESS intrinsic. Programmers that are familiar with this intrinsic will be able to create an FTP user process easily. No special entry point will be available for programmatic access. The FTP commands covered in the preceding section will be the same. If STDIN is redirected from the default, it must be an unnumbered file. It is the user's responsibility to certify that the commands are valid. All error messages and FTP server replies to commands will be directed to the redirected STDLIST or to the default. Break will be the same as above. Control-Y will only be enabled if STDIN and STDLIST have not been redirected. An "FTPJCW" will be set so the user can programmatically check if the transfer was successful.

3.1.3 File System Interface

The main discussion of the file system interface will be left to the Internal Design document, but some issues, which will affect the FTP user, should be addressed. The FTP Mil-Std has several limitations with which we must deal as well as possible. Some decisions were made which affect the user interface. At one time it was considered necessary to include a STRUCTURE command so the user could signify the source file structure. The parameters for this command are a) File - which is basically an unstructured file which is considered a continuous sequence of data, and b) Record - in which the data is broken into sequential records. Upon re-examination it was determined that the FTP user process should always set it to reflect the file structure of the 3000, that is "record." That will notify the server that the user file system is based upon a record structure rather than a "file" structure. The conversion to that structure will have to be done by the server process whether it be a 3000 or not.

File and record size may also be unknown before the transfer starts. If the source file is on the user 3000 system, an "ALLO" FTP command will be sent to let the server know the total file size and the size of each record. Unfortunately there is no mechanism to notify the user process, if the transfer is in the opposite

direction. If the file and record size are not known before the transfer, there will be arbitrary limits set for those parameters when the file is created. Basically those limitations will be the defaults provided by the MPE file system, except 1) ASCII will assume 256 bytes in variable-length records and 2) binary will be build with 128 words in variable-length records. They will be demonstrated in the examples at the end of this discussion. These limits will be well documented. The user will be able to override those limits by using the MPE BUILD command before the transfer. In the letter of the Mil-Std, the file will be overwritten if it already exists. In this manner the user can change the default file options.

The TYPE command has been left for the user; ASCII is the default. Binary, Ebcidic, and local byte size (logical byte which is actually indicative of the "word" size) are the other parameters. ASCII and binary will be supported. If the file will be a binary file, the user can use the command to notify the file system through the FTP subsystem. ASCII and BINARY will both build a file with filecode=0. If this is not satisfactory, the user must be encouraged to use the BUILD command to his own specifications and copy the file into that file. In the attempt to provide a generic file system interface in the standards, FTP provides us with little choice as to the file system interface.

3.1.4 Foreign Language Input

The command set keywords, delimiters, messages, etc. will be loaded from a well-known catalog file. This will probably be a part of a DDNCAT file for all DDN Services. This will provide the Network Manager the ability to localize these items.

3.2 SECURITY SPECIFICATIONS

Users of FTP will not be exempt from the usual MPE and file system security restrictions. Both local and remote users must logon to a valid account on the 3000. No special capabilities will be necessary or given to the user. FTP remote users can only use the FTP process. It will not be possible to break the remote session and thereby communicate to the remote CI process. The CI process will only serve as silent parent of the server and will be terminated after the local user terminates the connection.

3.3 INSTALLATION INSTRUCTIONS

This product should be released in the normal MIT cycle. The IND Spec 150 standards for file naming conventions, separate installation files, and MIT submittals will be strictly observed. All installation procedures will be thoroughly tested before MR of the product.

3.4 PERFORMANCE PREDICTIONS

FTP performance will be comparable to other implementations on comparable processors. An example of that performance was included in DDN Services/HP3000: Investigation Report. Also the performance should be comparable to that of DSCOPY in interchange mode operating within the same environment.

3.5 RELIABILITY AND RECOVERY

This product will be tested thoroughly for reliability. It must be determined that this product can run continuously for a minimum of 120 hours without critical or serious bugs, with an acceptable amount of low and normal bugs, and no system failures.

There is no data reliability check at this level of the DDN product. FTP assumes a reliable transport, i.e., all errors detected and either corrected or reported. In case the remote system crashes the user may or may not be notified by the transport layer that the connection has been lost. In the case where the user is not notified of the remote failure, the Control-Y mechanism discussed above will give the user the ability to check the status of a data transfer. If the user suspects that the data and/or command connection have been lost, s/he can check on the progress of the transfer by using the Control-Y. If necessary the user can abort the transfer via Control-Y. Connection lost during file transfer may result in unreliable or incomplete data. The user would be well-advised to restart any file transfer that was in progress when the error was detected.

3.6 ERROR MESSAGES

Any critical errors that are encountered will be logged as NS servers do. These errors may occur in any of the following categories. The errors that can be encountered are categorized into three types.

3.6.1 FTP User Process

Any error that is encountered during the operation of the FTP program will be reported to the user in a clear, meaningful manner. Some examples are "Unknown command," "Unknown host <host name>," etc. The action taken will be to return the user the prompt for continuation; it is the responsibility of the user to determine if the error is serious enough to warrant exiting the program.

3.6.2 Other Local Errors

All non-FTP errors reported during the running of the program will be reported to the user via the normal error messages generated by the operating system or other subsystems. This includes MPE errors and file system errors, as well as transport and X.25 errors (e.g., file system error "NON-EXISTENT FILE. (FSERR 52)" will be returned if a file specified does not exist on the local system).

3.6.3 Remote Server Errors

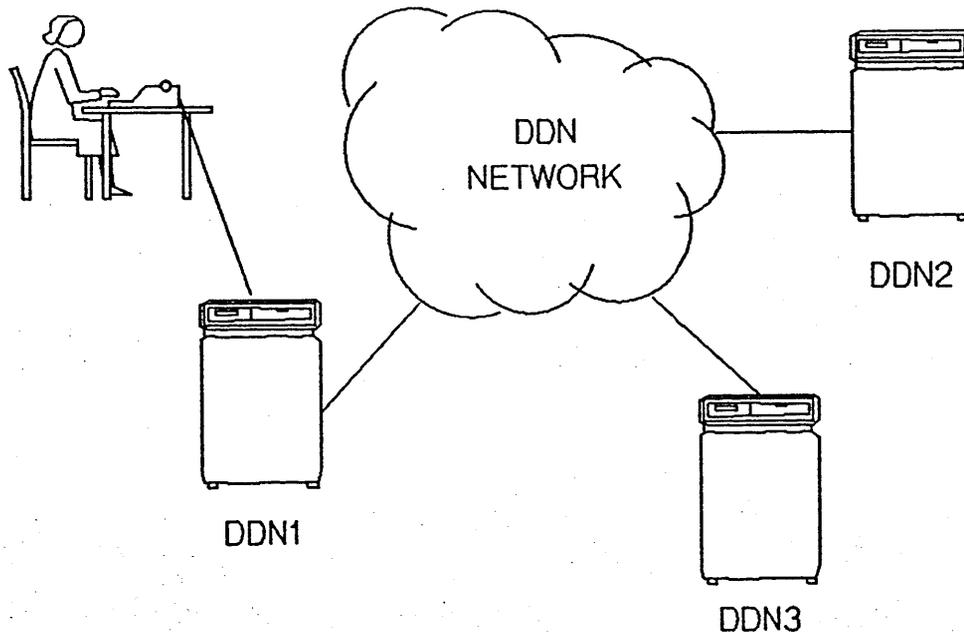
The FTP Mil-Std includes reply numbers and explanations of what the numbers mean. Since these are a part of the standard the numbers will be used. In line with other implementations, however, accompanying messages will reflect the usual error messages generated by the 3000, if an error condition exists. If there is no correlating error message for the condition, the server will return a clear, meaningful message to the user; usually the example in the standard. For example, an error accessing a non-existent file on the remote system will generate the error "550 NON-EXISTENT FILE. (FSERR 52)." It is assumed that the messages generated by the 3000 are clear and precise enough that non-HP users will be able to understand them. Some of the reply codes do not imply an error condition, but indicate a preliminary or intermediate condition. Some of these will be given to the user and some of them will be used solely for internal state transitions.

3.6.4 FTP Error Set

The following error numbers and the following text are taken directly from Military Standard, FTP, Mil-Std-1780. The text may be altered as outlined in 3.6.3 above. The type of reply is generally based upon the first digit of the code. 1yz indicates a positive preliminary reply; 2yz is a positive completion reply; 3yz is a positive intermediate reply; 4yz implies a transient negative reply which the user may wish to retry; and 5yz is a permanent negative reply.

- 110 Restart marker reply.
- 119 Terminal not available, will try mailbox.
- 120 Service ready in ! minutes.
- 125 Data connection already open; transfer starting.
- 150 File status okay; about to open data connection.
- 151 User not local; Will forward to !@!
- 152 User unknown; Mail will be forwarded by operator.
- 200 Command okay.
- 202 Command not implemented, superfluous at this site.
- 211 System status, or system help reply.
- 212 Directory status.
- 213 File status.
- 214 Help message.
- 215 ! is the preferred scheme.
- 220 Service ready for new user.
- 221 Service closing TELNET connection.
- 225 Data connection open; no transfer in progress.
- 226 Closing data connection; requested file action successful.
- 227 Entering Passive Mode. h1,h2,h3,h4,p1,p2
- 230 User logged in, proceed.
- 250 Requested file action okay, completed.
- 331 User name okay, need password.
- 332 Need account for login.
- 350 Requested file action pending further information.
- 354 Start mail input; end with <CR><LF>. <CR><LF>.
- 421 Service not available, closing TELNET connection.
- 425 Can't open data connection.
- 426 Connection closed; transfer aborted.
- 450 Requested file action not taken: file unavailable.
- 451 Requested action aborted: local error in processing.
- 452 Requested action not taken: insufficient storage space in system.
- 500 Syntax error, command unrecognized.
- 501 Syntax error in parameters or arguments.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.
- 530 Not logged in.
- 532 Need account for storing files.
- 550 Requested action not taken: file unavailable.
- 551 Requested action aborted: page type unknown.
- 552 Requested file action aborted: exceeded storage allocation.
- 553 Requested action not taken: file name not allowed.

3.7 FTP EXAMPLES



1) RUN FTP.NET.SYS;INFO="DDN2"

A user on DDN1 has invoked the FTP program on host DDN1. The program will establish a connection to DDN2 after it has found its address in the Network Directory. The remote server will have been given the connection by DSDAD and be waiting for the first command.

2) USER DAVID.DDN

The user logon string will be transferred to the server at DDN2 which will in turn use it to establish a session on that host. If a user password were necessary the message:

331 ENTER USER PASSWORD

would be sent back to the user, the local program would issue a password prompt, and would transfer the password to DDN2. If the logon were successful, the reply:

230 USER LOGGED ON

would be sent to the user as well as other any other messages that the remote server returns.

3) PUT FILE1

The source file FILE1 on the local DDN1 will be transferred to the remote DDN2 host. It will have the same name on the remote host as it does on the local.

4) PUT FILE1 RFILE1

As number 3) above, the local file, FILE1, will be transferred to DDN2 where it will have the name RFILE1.

5) GET RFILE1 FILE2

The file, RFILE1, on the remote host will be transferred back to the local machine. If the target file name had not been specified, it would have had the same name on the local system.

6) CLOSE

The remote session will be terminated and the connection will be closed to DDN2, but the FTP program will not terminate.

7) OPEN DDN3

Functional Specifications

A connection will now be opened to remote host, DDN3, where a server process will be started to handle commands from the local FTP user program.

8) **USER DAVID.DDN**

The user logon string will be transferred to host DDN3. The same sequence occurs as above.

9) **LISTF FILE1**

Since the DDN3 connection is now open the command will be routed to that server process. The server will transfer the requested listing to the local user.

10) **EXIT**

The connection to DDN3 will be closed and the remote session, therefore, will be logged off. The FTP program will terminate and the user will be returned to the MPE prompt.

Table of Contents

Section 1

PRODUCT IDENTIFICATION

Section 2

PRODUCT SPECIFICATIONS

2.1 Product Overview	2-1
2.2 User Definitions	2-1
2.2.1 Local User	2-1
2.2.2 Remote User	2-1
2.3 Product Environment	2-1
2.4 User Documentation Requirements	2-2

Section 3

FUNCTIONAL SPECIFICATIONS

3.1 User Interface	3-1
3.1.1 Interactive Interface	3-1
3.1.1.1 User Command Set	3-1
3.1.1.2 Break and Control-Y	3-3
3.1.2 Programmatic Interface	3-3
3.1.3 File System Interface	3-3
3.1.4 Foreign Language Input	3-4
3.2 Security Specifications	3-4
3.3 Installation Instructions	3-4
3.4 Performance Predictions	3-4
3.5 Reliability and Recovery	3-5
3.6 Error Messages	3-5
3.6.1 FTP User Process	3-5
3.6.2 Other Local Errors	3-5
3.6.3 Remote Server Errors	3-5
3.6.4 FTP Error Set	3-6
3.7 FTP Examples	3-7

**** END OF FORMATTING ****
TDP/3000 (A.03.11) HP36578 Formatter
MON, FEB 10, 1986, 6:14 PM
NO ERRORS
INPUT = ES.FTP.DDN
OUTPUT = *HP2680

#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM

#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM

#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM

#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM
#J174; #O2613 * ES, DAVID.DDN; SLP * MON, FEB 10, 1986, 7:10 PM