

NBSIR 84-2970

# File Transfer Over the NBSNET: User's Manual

---

Debra L. Strawbridge, Carol Young

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
National Engineering Laboratory  
Center for Applied Mathematics  
Gaithersburg, MD 20899

November 1984



---

U.S. DEPARTMENT OF COMMERCE  
NATIONAL BUREAU OF STANDARDS



NBSIR 84-2970

**FILE TRANSFER OVER THE NBSNET:  
USER'S MANUAL**

---

Debra L. Strawbridge, Carol Young

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
National Engineering Laboratory  
Center for Applied Mathematics  
Gaithersburg, MD 20899

November 1984

**U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary***  
**NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director***



FILE TRANSFER OVER THE NBSNET;

A USER'S MANUAL

by

Debra Strawbridge and Carol Young

ABSTRACT

File Transfer Protocol (FTP) is a set of specifications for the transfer of information between computers via the NBS local area network. A description is given for the implementations of the File Transfer Protocol for Perkin Elmer\* 32-bit and 16-bit computers and for the SPERRY 1100 series computers. User instruction and reference documentation are given for both the file transfer initiating programs and the remote host-interface programs. Sample file transfer sessions, description of error messages and summary of commands are included.

\* Certain trade names and company products are mentioned in the text. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the products are necessarily the best available for the purpose.

## TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION	5
2. FTP32	6
2.1 Introduction	6
2.2 Logical Units Assignments in OS/32	6
2.3 Starting FTP32	7
2.4 Command Mode	7
2.5 Transaction Commands	8
2.5.1 LOGIN	8
2.5.2 REPLACE, CREATE, APPEND	8
2.5.3 FETCH, DIRLIST	9
2.5.4 DELETE, SUBMIT, RENAME	9
2.5.5 DEFAULT	10
2.6 Control Commands	10
2.6.1 TELL, ASK	10
2.6.2 CANCEL	10
2.6.3 LOGOUT, QUIESCE, DISCONNECT, END	11
2.7 Local Commands	11
2.7.1 PAUSE	11
2.7.2 MODE	11
2.7.3 STATUS	12
2.8 Summary of Commands	12
2.9 Example	12
3. FTP16	14
3.1 Introduction	14
3.2 Starting FTP16	14
3.3 Command mode	14
3.4 Transaction Commands	15
3.4.1 LOGIN	15
3.4.2 REPLACE, CREATE, APPEND	16
3.4.3 FETCH, DIRLIST	16
3.4.4 DELETE, SUBMIT, RENAME	17
3.4.5 DEFAULT	17

	Page
3.5 Control Commands	18
3.5.1 TELL, ASK	18
3.5.2 CANCEL	18
3.5.3 END, DISCONNECT	18
3.6 Local Commands	19
3.6.1 LIST	19
3.6.2 DEVICE	19
3.6.3 RECORD	19
3.6.4 MODE	19
3.6.5 INPUT	20
3.6.6 STATUS	20
3.6.7 DO	20
3.6.8 PAUSE	21
3.7 Special Note	21
3.8 Summary of Commands	21
3.9 Description of Command Messages	22
3.10 Equipment Configuration	23
3.11 Logical Unit Assignments	23
3.12 Example	23
4. SERV32	26
4.1 Introduction	26
4.2 Transactions Commands	26
4.2.1 LOGIN	26
4.2.2 DEFAULT, VOLUME	27
4.2.3 DEFAULT MODE	27
4.2.4 REPLACE, CREATE, APPEND	27
4.2.5 FETCH	28
4.2.6 DELETE	28
4.2.7 RENAME	28
4.2.8 DIRLIST	28
4.3 Control Commands	29
4.3.1 TELL, ASK	29
4.3.2 CANCEL	29
4.3.3 LOGOUT	29
4.4 Unsupported Transactions	30
4.5 Error Responses	30
4.6 File Descriptors	30
5. SERV16	32
5.1 Introduction	32
5.2 Transaction	32
5.2.1 LOGIN	32

	Page
5.2.2 DEFAULT DEV	32
5.2.3 DEFAULT MOD	33
5.2.4 DEFAULT	33
5.2.5 REPLACE, CREATE	33
5.2.6 FETCH, DIRLIST	33
5.2.7 DELETE	34
5.3 Control Commands	34
5.3.1 TELL	34
5.3.1 ASK	34
5.3.2 CANCEL	34
5.3.3 END	35
5.4 Unsupported Transactions	35
5.5 Error Responses	35
5.6 File Descriptors	35
6. SERV1100	37
6.1 Introduction	37
6.2 Transaction Commands	38
6.2.1 LOGIN	38
6.2.2 DEFAULT	38
6.2.3 REPLACE, CREATE	38
6.2.4 FETCH	38
6.2.5 DELETE	39
6.2.6 SUBMIT	39
6.2.7 DIRLIST	39
6.3 Control Commands	39
6.3.1 TELL	39
6.3.2 CANCEL	39
6.3.3 LOGOUT	40
6.4 Unsupported Transactons	40
6.5 Error Responses	40
6.6 File and Element Descriptors	40



# FILE TRANSFER OVER THE NBSNET

## USER'S MANUAL

### 1. INTRODUCTION

File Transfer Protocol (FTP) is a set of specifications which has been implemented on computers at NBS to allow file transfer between these computers via the NBSNET. File transfer involves a user FTP program running on one host and a server FTP program running on the second host. The term 'user' or 'user program' is used to identify the computer program executed by an operator to instigate file transfers. The term 'server' or 'server program' is used to identify the computer program which receives file transfer commands from a user program via the NBSNET.

The commands issued by the operator of a user program are independent of the destination server. However, the arguments to these commands such as file specification or login parameters, are server dependent.

FTP commands are independent of the distant computer/operating system. The user need not know any of the job control language of the distant computer. The user need only know the distant computer's login parameters and the filename syntax. FTP commands include:

1. Transaction commands which allow the user to initiate a file transfer or a file manipulation with the remote computer.
2. Control commands which allow the user to initiate and terminate a transaction or a file transfer session, or to communicate to the computer operator of the remote facility.
3. Local commands which allow the user to control the interface with the local computer.

This document includes instructions for the use of a user program (FTP32) and a server program (SERV32) running on 32-bit Perkin-Elmer computers under OS/32, a user program (FTP16) and a server program (SERV16) running on 16-bit Perkin-Elmer computers under FLOXOS and a server program (SERV1100) running on a SPERRY 1100 series computer.

## 2. PERKIN-ELMER 32 NBSNET FILE TRANSFER USER PROGRAM

## FTP32

## 2.1 INTRODUCTION

The purpose of FTP32 is to initiate the transfer of files between Perkin-Elmer 32-bit computers and other computers via the NBSNET. No prejudice exist as to the nature of these files. They can be symbolic files, job stream files or binary files. FTP32 is a user program and as such always initiates a connection with a FTP server. FTP32 runs on Perkin-Elmer 32-bit computers equipped with network interface equipment and an operating system configured with network ports.

Some features of FTP32 are:

1. Transmits full 8-bit binary or ASCII.
2. Automatically assigns the next available local network port.
3. Prompts the user for the distant computer's address and makes the connection.
4. Supports a dialogue with the distant computer, guiding the user through the distant computer's login procedure.
5. Supports requests for data transfer transactions allowing the user to replace an existing file, create a new file, append to an existing file or fetch an existing file.
6. Supports requests for utility service transactions such as delete a file, submit a job stream, rename a file, or specify defaults for fields in the server file specification.
7. Permits messages to be directed to the console operator of the distant computer.
8. Permits the user to cancel a transaction or to request a disconnect upon completion of a transaction.
9. Informs the user of the distant computer's acceptance or rejection of a transaction, together with the reason for a rejection.
10. Permits the user to write into a new local file or to replace or append to an existing local file. When replacing an existing local file, the original file is not changed unless the transaction completes successfully.

## 2.2 LOGICAL UNITS ASSIGNMENTS IN OS/32

LU 3 Network port  
LU 5 Command input  
LU 6 Response output

FTP32 honors any assignments made by the user. Otherwise FTP32 assigns command input and response output to the console device, C:, and attempts to assign the network logical unit to a free network port. The network ports must be configured as FTPx:, where 'x' is a digit from 0 thru 9. FTP32 terminates if no network ports are available.

## 2.3 STARTING FTP32

The user can type 'FTP32' to invoke the standard CSS file, which simply loads and starts FTP32. The program attempts to make the standard logical unit assignments and if successful queries 'CONNECT'. The user responds with the network address of the distant computer. The program attempts to connect to the distant computer and if successful enters command mode. If the connection is not successful, the program again queries 'CONNECT' permitting the user to request another address. Responding to this request with an empty line terminates FTP32.

## 2.4 COMMAND MODE

The user can now initiate a transaction with the distant computer. Most host computers will require a login transaction and will reject all other transactions until the login is complete. FTP32 permits only one transaction to be active at a time. The user can use the CANCEL command to terminate a transaction prematurely.

The user requests a transaction with the distant computer, not the local Perkin-Elmer. For instance, if the user enters 'REPLACE DATAFILE', he is requesting the distant computer to replace its file named DATAFILE with the incoming data sent from the user's local file. If the user requests 'FETCH DATAFILE', he is requesting the distant computer to fetch file DATAFILE from its file system and send it to the user's file system. FTP32 queries the user for all local file information needed. When FTP32 has all the information it needs to complete its part of the transaction, it sends the transaction to the distant computer and waits for an answer.

If the answer is an accept, the user is informed and, if data transfer has been requested, the transfer proceeds. The user is informed when the data transfer is completed. For transactions which do not request data transfer, an accept completes the transaction.

If the answer is a reject, the user is informed as to the reason for the rejection and the transaction is terminated.

Once the data transfer has been started, the transfer can be aborted by:

1. The user by entering a CANCEL command.
2. FTP32 when it can no longer read or write to its local file due to I/O errors. The user is informed of the I/O error and the logical unit on which the error occurred.
3. The distant computer's FTP server, probably due to an I/O error on transfer of data to its files. The server should inform the user of the reason for the transfer abort.

The format of a command is: `COMMAND ARG1,ARG2,....` A comma is used as a field separator in these examples but some servers may require different field separators. Check each server's documentation. In the following listing of commands, the abbreviation for each command is underlined. Commands can be entered in upper or lower case. The first three letters of each command are acceptable except for LOGIN and LOGOUT. Required arguments are capitalized.



FILENAME always refers to the distant computer's file and not the local file. The format of FILENAME is determined by the distant computer's FTP server. Refer to the distant computer's server documentation. Warning: Some commands implemented in FTP32 may not be implemented in the distant computer's FTP server.

## 2.5 TRANSACTIONS COMMANDS

These commands can only be entered when there is no transaction active.

### 2.5.1. Command to satisfy login requirements of the distant computer:

LOGIN access-parameter,access-parameter,...

The simplest way to sign on to a distant computer is to type LOGIN with no arguments. The distant FTP server will prompt for access parameters needed. If the user knows the access parameters needed, he can send them as arguments with the LOGIN command. The server will prompt for any missing access parameters. The user is informed when the LOGIN has been accepted and most FTP servers will send an indentifying signon line.

### 2.5.2. Commands which initiate data flow from the local Perkin-Elmer to the distant computer:

REPLACE FILENAME  
CREATE FILENAME  
APPEND FILENAME

FILENAME represents the name of the distant computer's target file. The REPLACE and APPEND commands request the distant FTP server to use FILENAME as the target file if it exists, or to create a new file if it does not exist. The CREATE command requests the distant FTP server to create a new file and to reject the transaction if the file already exists.

REPLACE and CREATE commands are used to transfer files from the local Perkin-Elmer to the distant computer creating an exact copy of the local file in the target file.

APPEND command is used to transfer a local file and have it appended to the distant computer's file.

FTP32 requests the local file name and attempts to assign it to logical unit 1. If the file cannot be assigned, the user is told why and again queried for the file name. An empty answer to this request aborts the transaction.

The local file is closed when the transaction completes either normally or is aborted with a CANCEL, (see Section 2.6.2).

2.5.3. Commands which initiate data flow from the distant computer to the local Perkin-Elmer:

FETCH FILENAME  
DIRLIST FILENAME

FILENAME represents the name of the distant computer's file. The transaction is rejected if FILENAME does not exist.

The FETCH command is used to transfer a complete file from the distant computer's file to a local Perkin-Elmer file.

The DIRLIST command is used to transfer a directory listing from the distant computer to the local Perkin-Elmer.

FTP32 queries the user for the target local file.

1. When the local file does not exist, the user is asked for the record length, and a new index file is allocated and assigned to logical unit 2.
2. When the file does exist, the user is asked whether to delete and reallocate this file, or to append to this file.

If the user requests delete and reallocate, a scratch file is allocated and assigned to logical unit 2. If the transaction completes successfully, the requested file is deleted and the scratch file is renamed to the requested file. If the transaction does not complete successfully, the requested file is not deleted, the user is given the name of the scratch file and the transaction terminates. When the requested file is a contiguous file, FTP32 creates an indexed scratch file with a record length of 256.

If the user requests append, the file is assigned to logical unit 2 in exclusive write mode. The file is closed when the transaction completes either normally or is aborted with a CANCEL, (see Section 2.6.2). If the requested file is a contiguous file, the meaning of the append function changes to a rewrite function and the contiguous file is rewritten from the beginning. It is the responsibility of the user to insure that the contiguous file is large enough to receive all the incoming data.

2.5.4. Commands to request utility service from the distant computer:

DELETE FILENAME  
SUBMIT FILENAME  
RENAME FILENAME,NEWFILENAME

FILENAME represents the name of the distant computer's file. The transaction is rejected if FILENAME does not exist.

DELETE requests the removal of FILENAME from the distant computer's file system.

SUBMIT requests the submission of FILENAME as a batch run on the distant computer. FILENAME must contain a legal batch job runstream for the distant computer.

RENAME requests the distant computer to change the name of the file listed in the first argument to the name given in the second argument. NEWFILENAME cannot already exist in the distant computer file system. These transactions require no data transfer. The transaction is completed when it is accepted.

#### 2.5.5. Command to specify defaults for fields in the server's filename.

##### DEFAULT parameter

The DEFAULT command specifies a default field which the distant server uses to identify files requested in subsequent transactions. The exact meaning of this command is determined by the receiving server. The default field is usually some part of the filename specification common to all requested files, for example: directory name, qualifier or volume name. The user should check the appropriate server documentation for the DEFAULT meaning. Sending the DEFAULT command with an argument will cancel any previously specified default and set the current default to the value given in the argument. The distant computer's FTP server will now use this information in conjunction with filename specifications sent in upcoming commands to identify the requested file. Sending the DEFAULT command without an argument will cancel any previously specified default. The user now has to specify the complete distant file specification in upcoming commands or use the system default volume of the distant computer.

## 2.6 CONTROL COMMANDS

These commands are not transactions and can be entered at any time.

### 2.6.1. Commands to allow the user to send messages to the distant computer's console:

TELL MESSAGE  
ASK MESSAGE

MESSAGE represents the text the user wants to send to the distant computer's console.

TELL sends the message to the distant computer's console.

ASK sends the message to the distant computer's console and requests an answer.

### 2.6.2. Command to abort an active transaction:

CANCEL

If the distant computer is transmitting data, FTP32 sends a CANCEL request to the distant computer and discards any further data received. The receiving local file is closed. If the user had specified the delete and



reallocate option, the requested file is not deleted. The user is given the name of the scratch file containing the data transferred before the transaction was aborted.

If the local computer is transmitting data, FTP32 stops sending the data and sends an ABORT request to the distant computer.

If the transaction is a LOGIN or a transaction involving no data transfer, the transaction is aborted locally only.

If no transaction is active, the CANCEL command is ignored.

2.6.3. Commands to terminate the connection with the distant computer FTP server:

```
LOGOUT
QUIESCE
DISCONNECT
END
```

LOGOUT sends a terminate command to the distant computer's FTP server. The physical connection is not broken and FTP32 remains in command mode. A user can type LOGIN to initiate a new file transfer session with the distant server.

QUIESCE, DISCONNECT and END send a terminate command to the distant computer's FTP server and a disconnect function to the network. FTP32 goes to end of task with normal return code of 0.

If a transaction is active when a DISCONNECT or END command is received, the transaction is cancelled before the disconnect action. However an active transaction is allowed to go to completion before the disconnect action when a QUIESCE command is received. After receiving a QUIESCE command, FTP32 will not accept further commands from the user.

## 2.7 LOCAL COMMANDS

2.7.1. Command to allow the user to return to operating system control, temporarily:

```
PAUSE
```

To return to FTP32, the user issues the CONTINUE command. Upon re-entering FTP32, the user is in command mode and any active transaction continues.

2.7.2. Command to set ASCII or BINARY I/O with the local file manager.

```
MODE ASCII
MODE binary
```

This command only affects the local computer. Prior to a MODE command FTP32 requests ASCII I/O from the local file manager. The user can use the MODE command to change to binary I/O and back to ASCII I/O. When a MODE command is

entered without an argument, FTP32 reports the current mode.

2.7.3. Command to print the status of the network connection:

STATUS

Prints on the user's consol the address of the local and distant network ports.

2.8 SUMMARY OF COMMANDS

LOGIN access parameter,access parameter.....

REPLACE FILENAME

CREATE FILENAME

APPEND FILENAME

FETCH FILENAME

DIRLIST FILENAME

DELETE FILENAME

SUBMIT FILENAME

RENAME FILENAME,NEWFILENAME

DEFAULT parameter

TELL MESSAGE

ASK MESSAGE

CANCEL

LOGOUT

QUIESCE

DISCONNECT

END

PAUSE

MODE

STATUS

2.9 EXAMPLE

The following is an example to a FTP session with SERV1100 on a SPERRY 1100 series computer computer. The command typed by the FTP32 user are underlined.

FTP32

CONNECT?

1108

CONNECTED

LOGIN USER,12345-USERID,PROJECT-ID

LOGIN ACCEPTED

SPERRY NBSNET SERV1100

REPLACE DATAFILE.

ENTER LOCAL FILE NAME

DATAFILE.DAT

TRANSACTION ACCEPTED

TRANSMISSION IN PROGRESS

TRANSMISSION COMPLETE

END OF FILE



FETCH PROGRAM.

ENTER LOCAL FILE NAME (EMPTY LINE WILL CANCEL TRANSACTION)

PROGRAM.FTN

NEW FILE ENTER RECORD LENGTH OF PROGRAM.FTN

80

TRANSACTION REJECTED

FILE DOES NOT EXIST

FETCH PROGRAM1.

ENTER LOCAL FILE NAME (EMPTY LINE WILL CANCEL TRANSACTION)

PROGRAM.FTN

DELETE AND REALLOCATE OR APPEND TO PROGRAM.FTN

D

TRANSACTION ACCEPTED

TRANSMISSION IN PROGRESS

END OF FILE, TRANSMISSION COMPLETE

END

FTP32 END

### 3. PERKIN-ELMER 16 NBSNET FILE TRANSFER USER PROGRAM

#### FTP16

##### 3.1 INTRODUCTION

The purpose of FTP16 is to initiate the transfer of files between a Perkin-Elmer 16-bit computer and other computers via the NBSNET. FTP16 runs under the FLOXOS Operating System. The files that are transferred can be either ASCII or binary files. FTP16 is a user program and initiates a file transfer session with a FTP server program running on a distant computer.

##### 3.2 STARTING FTP16

To start the FTP user program, the user enters < RU FTP16 >. The program enters an initialization phase where the logical unit assignments are made. Refer to section 3.11 for a description of these assignments. If an error occurs during the initialization phase, FTP16 terminates. Otherwise, once the phase is completed, the user is queried for a connect address. The user responds with the network address of the distant computer. The program attempts to connect to the distant computer and, if successful, enters command mode. If the connection is unsuccessful, the program again queries the user for a connect address. Responding to this request with an empty line terminates FTP16.

##### 3.3 COMMAND MODE

The user can now initiate a transaction with the distant computer. Most host computers will require a login transaction. This is used to identify the user to the distant computer. All other transaction commands will be rejected until the login transaction is complete. After the login is complete, the user can enter any transaction command. FTP16 permits only one transaction to be active at a time.

The user requests a transaction with the distant computer, not the local Perkin-Elmer. FTP16 queries the user for all local file information needed and checks to make sure all local file information is valid. If it is not, the user is told the reason and the command is rejected locally. If the information is valid, FTP16 sends the transaction to the distant computer and waits for an answer.

If the answer is an accept, the user is informed and, if data transfer has been requested, the transfer proceeds. The user is informed when the data transfer is completed. For transactions which do not request data transfer, an accept completes the transaction.

If the answer is a reject, the user is informed as to the reason for the rejection and the transaction is terminated.

FTP16 prints all rejects and accepts to the console to notify the user of the state of the transaction or command. Refer to section 3.9 for a description of these messages.

Commands supported by FTP16 are either transaction commands which are issued to the distant computer or local commands which are issued to the local computer. Commands such as initiating a file transfer, setting the mode of a transaction or sending a message to the distant computer's console are examples of transaction commands. Commands setting the local defaults or listing the directory of a local disk are examples of local commands.

The following sections describe the commands supported by FTP16. Arguments in lower case are optional. The first three characters of each command will be accepted as an abbreviation of the command. FILENAME refers to the distant computer's file and not the local file. File information for the distant computer can be found in this document under the section for the appropriate server. Local filename refers to any valid FLOXOS filename. This consists of up to six printable ASCII characters. All printable ASCII characters are valid. The maximum record length for a file supported by FTP16 is 256 characters. This parameter is set during the initialization phase. Other file parameters are set during the initialization phase and are used by some commands as default values. They are as follows:

Physical device address - current default physical device  
 Record length - 80 characters per record  
 Character mode - ASCII  
 Input mode - local file mode

The description of each command will specify when these default parameters will be used and how.

It should be noted that some commands supported by FTP16 may not necessarily be implemented in the distant computer's FTP server. For information on a specific FTP server refer to the section for that server in this document.

### 3.4 TRANSACTION COMMANDS

These commands are issued to the distant computer. They should only be entered when there is no transaction active. If one of these commands is entered while another transaction is active, the command being requested will be rejected.

#### 3.4.1. Transaction to satisfy login requirements of the distant computer:

LOGIN access-parameter,access-parameter,...

The simplest way to sign on to a distant computer is to type LOGIN with no arguments. The distant FTP server will prompt for access parameters needed. If the user knows the access parameters needed, he can send them as arguments with the LOGIN command. The server will prompt for any missing access parameters. The user is informed when the LOGIN has been accepted and most FTP servers will send an identifying signon line.



3.4.2. Transactions which initiate data flow from the local Perkin-Elmer to the distant computer:

REPLACE FILENAME  
CREATE FILENAME  
APPEND FILENAME

FILENAME represents the name of the distant computer's target file. The REPLACE and APPEND commands request the distant FTP server to use FILENAME as the target file, if it exists, or to create a new file if it does not exist. The CREATE command requests the distant FTP server to create a new file and to reject the transaction if the file already exists.

The REPLACE and CREATE commands are used to transfer files from the local Perkin-Elmer to the distant computer, creating an exact copy of the local file in the target file.

The APPEND command is used to transfer a local file and have it appended to the distant computer's file.

FTP16 requests the local file name and its physical device address, and attempts to assign it to logical unit 1. If the file cannot be assigned, the user is told why and the command is rejected. A carriage-return in response to the question for the local file will terminate the command. The default physical device will be used if a carriage return is entered in response to the question for the physical device address. If the file assignment is successful, the user is queried whether to send until an end-of-file is encountered or to send the number of records specified in the directory. This is an attempt to avoid the problems that arise when a filemark has not been written to the file. After this question, the transaction command is sent to the distant computer. If it is accepted, the transfer begins. If it is rejected, the user is told why and the command is rejected.

The local file is closed when the transaction completes either normally or is aborted with a CANCEL command.

3.4.3. Transactions which initiate data flow from the distant computer to the local Perkin-Elmer:

FETCH FILENAME  
DIRLIST FILENAME

FILENAME represents the name of the distant computer's file or directory. The transaction is rejected if FILENAME does not exist.

The FETCH command is used to transfer a complete file from the distant computer's file to a local FLOXOS file.

The DIRLIST command is used to transfer a directory listing from the distant computer to the local Perkin-Elmer. In some cases, FILENAME on this command may be optional. The user should check the appropriate server documentation for the syntax of this command.

FTP16 queries the user for the target local file and its physical device address. If the local file does not exist, FTP16 queries the user about whether to proceed and allocate the file or to cancel the command. If the user chooses to continue, FTP16 queries the user for the record length and number of cylinders for the file. A carriage-return to the question for the record length will cause the default record length to be used. One cylinder will be allocated if a carriage-return is the response to the question for the number of cylinders. FTP16 attempts to allocate the file and, if successful, assigns it to logical unit 2. If unable to allocate the file, the user is told why and the command is rejected.

To protect the user from overwriting a file by mistake, if the file already exists, the user is asked whether to proceed with the command. The file will be overwritten if YES is the response. A NO will cancel the command.

It is possible to send the input to the console instead of a local file. This is useful when listing the directory of the distant computer. Refer to the documentation on the INPUT command under the section 3.6.5.

#### 3.4.4. Transactions to request utility service from the distant computer:

DELETE FILENAME  
SUBMIT FILENAME  
RENAME FILENAME,NEWFILENAME

FILENAME represents the name of the distant computer's file. The transaction is rejected if FILENAME does not exist.

DELETE requests the removal of FILENAME from the distant computer's file system.

SUBMIT requests the submission of FILENAME as a batch run on the distant computer. FILENAME must exist on the distant computer and contain a legal batch job runstream for the distant computer.

RENAME requests the distant computer to change the name of the file listed in the first argument to the name given in the second argument. NEWFILENAME cannot already exist in the distant computer file system.

These transactions require no data transfer. The transaction is completed when it is accepted.

#### 3.4.5. Transaction to specify defaults for the distant computer:

DEFAULT argument

The exact meaning of this command is determined by the distant computer's server. The default field can be some part of the filename specification common to all requested files or specify a certain mode the program is to run in, such as ASCII or binary. The command specifies a default field which the distant server uses in subsequent transactions until another DEFAULT command is received to reset the field. Refer to the appropriate server documentation

in this manual for the format of this command and its specific meaning to that server.

### 3.5 CONTROL COMMANDS

These commands are issued to the distant computer. They are not transactions and can be entered at anytime.

#### 3.5.1. Commands to allow the user to send messages to the distant computer's console:

TELL MESSAGE

ASK MESSAGE

MESSAGE represents the text the user wants to send to the distant computer's console.

TELL sends the message to the distant computer's console.

ASK sends the message to the distant computer's console and requests an answer. The user does not need to wait for an answer before initiating a transaction. The response to this request is a TELL command from the distant computer.

#### 3.5.2. Command to abort an active transaction:

CANCEL

If the distant computer is transmitting data, FTP16 sends a CANCEL request to the distant computer and discards any further data received. The receiving FLOXOS file is closed but no filemark is written to the file.

If the local computer is transmitting data, FTP16 stops sending the data and sends an ABORT request to the distant computer. The state of the file on the distant computer is dependent on the distant computer's server program. Refer to the appropriate server documentation in this manual.

If no transaction is active, the CANCEL command is ignored.

#### 3.5.3. Commands to terminate the connection with the distant computer's FTP server:

END

DISCONNECT

The END and DISCONNECT commands send a terminate command to the distant computer's FTP server and a disconnect function to the network. FTP16 goes to end-of-job.

If a transaction is active when a DISCONNECT command is received, the transaction is cancelled before the disconnect action. However, an active transaction is allowed to go to completion before the disconnect action when



an END command is received. After receiving an END command, FTP16 will not accept further commands from the user.

### 3.6. LOCAL COMMANDS

The following commands only affect the local computer and are not sent to the distant computer.

#### 3.6.1. Command to list on the console the directory for a physical device address (pp):

LIST pp

This command will list the local directory for physical device pp on the console if the physical device address specified is valid. If it is not valid, the command is rejected. If no physical device address is specified, the default physical device is used.

#### 3.6.2. Command to set FTP16's default physical device address:

DEVICE pp

This command will set FTP16's default physical device to pp if pp is a valid physical device address. If not, the command is rejected. Prior to a DEVICE command, FTP16 uses the default physical device set when the program is first initiated. The command with an asterisk as the argument sets the default back to the initial physical device. When a DEVICE command is entered without an argument, FTP16 reports the current default physical device.

#### 3.6.3. Command to set the default record length:

RECORD length

This command will set the default record length to the length specified if it is valid. If not, the command is rejected. Prior to a RECORD command, the default is set at 80. The command with an asterisk as the argument sets the default record length back to 80. When a RECORD command is entered without an argument, FTP16 reports the current record length.

#### 3.6.4. Command to set ASCII or binary I/O with the local file manager:

MODE A  
MODE B

This command will set the character mode to either ASCII (MODE A) or binary (MODE B). The default character mode is ASCII. In this mode, FTP16 requests ASCII I/O from the local file manager. The user can change to binary I/O and back to ASCII I/O using this command. The command entered with an asterisk as the argument sets the default mode to ASCII. FTP16 will report the current mode when the command is entered with no arguments.

- 3.6.5. Command to set the console as the input file for a transaction command which initiates data flow into the local computer:

INPUT C  
INPUT F

This command sets the default input device for transactions which initiate data flow into the local computer. The INPUT C command sets the input device to the console. The INPUT F command sets the input device to a local file. The default input mode is local file mode. In this mode a local file is requested for all data transfer. All incoming data is written to the file specified. If the user enters an INPUT command with the type 'C' and then follows that command with a transaction command initiating data flow to the local computer, FTP16 will not query the user for a local file but will write all data coming in directly to the console. Once an input mode is set, it remains the mode until another INPUT command is entered to change it. This command entered with an asterisk as an argument will set the default input mode to local file mode. If the command is entered with no arguments, FTP16 reports the current input mode.

This command is very useful when using the DIRLIST command to see a directory from the distant computer on the console rather than write it to a file.

- 3.6.6. Command to print the status of the network connection and the local default settings:

STATUS

This command prints to the console the address of the local and distant network ports, and the local default settings for the MODE, DEVICE and RECORD commands.

- 3.6.7. Command to set a local file as the source of command inputs:

DO LOCAL-FILENAME

This command can only be entered when no other transaction is active. It is similar to the DO command under FLOXOS. All input will be read from this file until a '/' is encountered or an end-of-file is read. Any command can be read from this file. If a command requires a local file, FTP16 expects the local file name to be the next record in the DO command file. The local file name must also specify a physical device address unless the default is desired. The format is FILE,pp. The local file must exist even if the transaction requested is for input of data. If the distant server needs to query for further information, the user will be prompted for that information. The format of the name of the file containing the FTP16 commands is as follows: FILE,pp. The file name must be specified and must exist. If no physical device address is specified, the default physical device is assumed.

The message DO COMMAND ACTIVATED is printed when the DO command is first initiated. When a command is read from the command file, it is printed on the



console to notify the user which command is currently being processed. The message DO COMMAND DEACTIVATED is printed when the end of the file is reached and the DO command is terminated. If some type of error occurs while processing a command from the DO file, that command will be cancelled and the next command in the DO file will be executed.

It is possible to cancel a DO command in progress. This is done by entering an < esc character >. This is to allow an exit in case there is a mistake in the DO command file.

3.6.8. Command to pause FTP temporarily and to return to the control of the operating system:

PAUSE

This command can be entered at any time and will return to the control of the operating system until the user issues the CONTINUE command. Upon re-entering FTP16, the user is in command mode and any active transaction continues.

### 3.7. SPECIAL NOTE

If the console or the system appear to be in an unusual state anytime during the FTP16 session, the user should attempt to clear the problem by entering an < esc >. This will either reset the read to the console and prompt the user again, or, if the system is hung, cancel everything and send a disconnect. In the latter case, FTP16 will go to end-of-job.

### 3.8. SUMMARY OF COMMANDS

LOGIN access parameter,access parameter ....  
 REPLACE FILENAME  
 CREATE FILENAME  
 APPEND FILENAME  
 FETCH FILENAME  
 DIRLIST FILENAME  
 DELETE FILENAME  
 SUBMIT FILENAME  
 RENAME FILENAME,NEWFILENAME  
 DEFAULT parameter  
 TELL MESSAGE  
 ASK MESSAGE  
 CANCEL  
 END  
 DISCONNECT  
 LIST pp  
 DEVICE pp  
 RECORD length  
 MODE parameter  
 INPUT parameter  
 STATUS  
 DO LOCAL FILE NAME  
 PAUSE

### 3.9. DESCRIPTION OF COMMAND MESSAGES

FTP16 prints all rejects and accepts to the console to notify the user of the state of the transaction or command. The following is a list of the standard messages printed.

There are two messages which are printed when a transaction is accepted and completed:

1. TRANSACTION ACCEPTED AND IN PROGRESS - this message is printed when a transaction, which initiates a file transfer, has been accepted. The transaction is in progress.
2. TRANSACTION COMPLETED - this message is printed when a transaction has completed. For those transactions, which do not initiate a file transfer, this message implies the transaction has been accepted and completed.

There are two types of reject messages printed:

1. COMMAND REJECTED - the command is never sent to the distant computer but rejected locally for a given reason.
2. TRANSACTION REJECTED - the command is sent to the distant computer and it is the distant computer which rejects the transaction.

For command and transaction rejects, the cause of the reject is either included as explanatory text in the reject message or one of the following messages:

NA-ERR	an invalid local file name was specified, an attempt was made to allocate a file that already existed, or to assign a file that did not exist.
LU-ERR	an invalid logical unit was specified and is not valid in FTP16.
PU-ERR	an invalid physical device address was specified.
CY-ERR	an invalid number of cylinders was specified or the number of cylinders did not exist contiguously on the specified disk pack.
RL-ERR	an invalid record length was specified.
DO-ERR	a directory overflow occurred. No other allocations can be made.
MO-ERR	invalid mode type specified.
I/O ERROR xxpp	an I/O error occurred when accessing a file or directory where xx is an error status and pp is a physical device address.

I/O error status codes are as follows:

X'CO'	a read or write operation was attempted on a file with a read or write protect attribute or a write operation was attempted on a file with hardware write protect enabled.
X'A0'	an attempt was made to use a physical device which was unavailable.
X'90'	an end-of-medium was reached on a file.
X'84'	a parity error was detected on a read operation or a defective track condition occurred.

### 3.10. EQUIPMENT CONFIGURATION

FTP16 runs on Perkin-Elmer 16-bit computers equipped with network interface equipment (TIE), RS-232C interface and a FLOXOS operating system configured with full duplex network ports. The baud rate is set at 9600.

### 3.11. LOGICAL UNIT ASSIGNMENTS

	PHYSICAL DEVICE ADDRESS
LU 1 - used for input file	
LU 2 - used for output file	
LU 3 - Network read	28
LU 4 - Network write	29
LU 5 - Console command input	20
LU 6 - Console command output	20
LU 7 - Precision interval clock	6C
LU 8 - used for DO command	

FTP16 attempts to assign logical unit 3, 4, 5, 6 & 7 upon starting the program. If there is a problem in assigning these logical units the program will terminate. If all assignments are completed successfully the program begins. Logical unit 1, 2 & 8 are only assigned when needed. If there is a problem in assigning these logical units, only the initiating command is rejected, the program does not terminate.

### 3.12. EXAMPLE

```

RU FTP16
ENTER CONNECT ADDRESS('CR' WILL BE TAKEN AS AN 'END')
>1108
CONNECTED - LOCAL: 2900           DISTANT: 1108
LOCAL DEFAULTS: DEV= DO MOD= A REC= 80
LOCAL COMMAND MODE
>LOGIN
ENTER USERID
>LEVIS
ENTER ACCOUNT NUMBER

```

```

>99999-111111
ENTER PROJECT-ID
>LEVIS
  TRANSACTION COMPLETED
  SPERRY NBSNET FTP.0181
>CREATE DATA.
ENTER LOCAL FILENAME
>FILMGR
ENTER DEVICE ADDRESS FOR LOCAL FILE
<CR>
SEND TO EOF(E) OR NUMBER OF RECORDS IN DIRECTORY(R)
(DEFAULT 'E')
>E
ENTER RECORD LENGTH FOR FILE DATA.
>80
  TRANSACTION ACCEPTED AND IN PROGRESS
  --another transaction command cannot be entered
>REPLACE FTPDOC.SMAIN
  TRANSACTION ALREADY ACTIVE
  --a non-transaction command can be entered
>STAT
  CONNECTED - LOCAL: 2900  DISTANT: 1108
  LOCAL DEFAULTS: DEV= D0 MOD= A REC= 80
> END OF FILE
  TRANSACTION COMPLETE
>FETCH DATA.
ENTER LOCAL FILE
>INTER
ENTER DEVICE ADDRESS FOR LOCAL FILE
>D1
  --The file INTER does not exist on disk D1
> NO SUCH FILE OR DIRECTORY INTER
DO YOU WISH TO ALLOCATE(Y OR N)
>Y
ENTER NUMBER OF CYLINDERS FOR LOCAL FILE
>10
ENTER RECORD LENGTH FOR INTER
>80
  TRANSACTION ACCEPTED AND IN PROGRESS
> END OF FILE
  TRANSACTION COMPLETED
>DELETE DATA.
TRANSACTION COMPLETED
>INPUT C
>INPUT
  C
  --at this point any data from a transaction which comes
  --into the Perkin-Elmer will be printed to the console
>DIR FTP.
  TRANSACTION ACCEPTED AND IN PROGRESS
  ELT  FTP16
  ELT  SERV16
> END OF FILE
  TRANSACTION COMPLETED

```

>END  
END OF FTP  
STOP  
EOJ



## 4. PERKIN-ELMER 32 NBSNET FILE TRANSFER SERVER

## SERV32

## 4.1 INTRODUCTION

SERV32 is a program that runs on Perkin-Elmer 32-bit computers and implements the NBSNET file transfer protocol. SERV32 never initiates a file transfer session but waits for a remote user to initiate a file transfer session with it. The remote user requests the file transfer transactions and SERV32 determines the validity of the transactions and executes valid transactions. Invalid transactions are rejected. A valid LOGIN transaction is required before any file transfer transactions are accepted. Only one user can be connected at a time. Other users will not be able to connect. One transaction can be active at a time.

Below is a list of transactions and other commands supported by SERV32. FILENAME represents the name of Perkin-Elmer file or Perkin-Elmer allocate specification. Comma is the field separator for all commands which have more than one argument. Abbreviations for the specification fields in the DEFAULT command are underlined. A description of error messages is included and the syntax of SERV32 file descriptor, FILENAME, is explained. For a more detailed explanation of each transaction, see the FTPuser Manual written for the user's FTP program.

## 4.2 TRANSACTIONS COMMANDS

## 4.2.1. Transaction to request access to SERV32:

```
LOGIN [userid],[account number],[password]
```

LOGIN is equivalent to SIGNON under Perkin-Elmer's Multi-Terminal-Monitor, MTM. SERV32 requires the same login parameters as MTM. These parameters do not need to be included on the login request. SERV32 will prompt for any missing parameters. USERID is the name by which the user wishes to be known to the system. Account number and password are verified with entries in an FTP accounting file and not with entries in the MTM accounting file. A user with a valid MTM account and password will not necessarily be able to login to SERV32. Check with the system staff of the computer to be accessed to determine file transfer privileges.

If the login parameters are valid, the LOGIN transaction is accepted and the user has read and write access to files catalogued under his FTP account, and read access to system files and his FTP group files. SERV32 sends a signon line identifying itself.

If the login parameters are not valid, the LOGIN transaction is rejected.

4.2.2. Transaction to allow the user to specify a default volume:

```
DEFAULT VOLUME=xxxx
```

SERV32 checks the syntax of the volume name, xxxx, and if correct accepts the DEFAULT transaction. This acceptance does not guarantee that the requested volume is available.

A DEFAULT VOLUME transaction without an argument cancels any previously set default volume.

When SERV32 receives a file descriptor in a transaction request, it determines the disk volume requested in the following order:

1. A volume specified in the file descriptor.
2. A volume previously specified in a DEFAULT transaction.
3. The default system volume.

4.2.3. Transaction to set I/O mode between the SERV32 and its file manager.

```
DEFAULT MODE=ASCII  
DEFAULT MODE=binary
```

Prior to a MODE setting, SERV32 is in the ASCII I/O mode. Upon receiving a DEFAULT MODE command, SERV32 sets the appropriate I/O mode with its file manager.

A DEFAULT MODE command without an argument reports the current mode.

4.2.4. Transactions to transfer information from the user's files to the server's files.

```
REPLACE FILENAME  
CREATE FILENAME  
APPEND FILENAME
```

With the REPLACE transaction, if FILENAME already exists, SERV32 accepts the transaction and transfers the incoming data into a scratch file. If the transaction completes successfully, the requested file is deleted and the scratch file is renamed to the requested file. If the rename fails, the user is informed and given the name of the scratch file. When the requested file is a contiguous file, SERV32 creates an indexed scratch file of record length 256.

With the CREATE transaction, if FILENAME already exists, SERV32 rejects the transaction.

With the APPEND transaction, if FILENAME already exists, SERV32 accepts the transaction and appends the incoming data to the existing file.

With all three transactions, if FILENAME does not exist, SERV32 creates a new file. The user can include in FILENAME any or all of the ALLOCATE command fields. SERV32 will substitute default values of INDEX, bsize =

1, isize = 1 and no keys if the corresponding fields are not specified. SERV32 does not default record or file size, and requests record size or file size from the user if not specified. Once the new file has been created, SERV32 accepts the transaction and writes incoming data into the new file.

The user should only create a contiguous file when sure of the file size. Otherwise it is safer to create an indexed file of record length 256 and later copy the indexed file into a contiguous file.

SERV32 allows users to write private files. If group or system file classes are specified in FILENAME, the transaction is rejected. If no file class is specified, private file class is assumed.

#### 4.2.5. Transaction to transfer information from the server's files to the user's files:

```
FETCH FILENAME
```

If FILENAME exists, SERV32 accepts the transaction and transfers the contents of this file to the user. If FILENAME does not exist, the transaction is rejected.

#### 4.2.6. Transaction to delete a file from the server's file system:

```
DELETE FILENAME
```

If FILENAME exists and is the user's private file, it is deleted and the transaction accepted. If FILENAME does not exist or is a group or system class file, the transaction is rejected.

#### 4.2.7. Transaction to change the name of a file in the server's file system:

```
RENAME OLDFILENAME,NEWFILENAME
```

If OLDFILENAME exists and is the user's private file, SERV32 attempts to rename it the NEWFILENAME. If successful the transaction is accepted. If the rename fails, the transaction is rejected and the user is given the rename command error status. If OLDFILENAME does not exist, or it is a group class file or a system class file, the transaction is rejected.

#### 4.2.8. Transaction to transfer a listing of the volume directory of a server's disk to a user file:

```
DIRLIST [voln:][filename][.ext][/acct]
```

The user can request a directory listing of all system files, all files under the user's account, or a subset of either set of files by specifying parameters similar to the display files command under Perkin-Elmer's OS/32.

The voln: field specifies the name of the requested disk volume. If this field is not specified, the user's default volume - if set by a previous 'DEFAULT VOL=' transaction - will be used. If no user default volume has been set, then the server's system default volume will be used.



The filename or extension field can be specified in full, or partially specified using a hyphen, to indicate that all files starting with the characters preceding the hyphen are to be listed subject to any restrictions in the other fields.

The acct field can specify either 'S' for system account, 'P' for user account, or '-' for system and user accounts.

For example:

PROG-.FTN lists files with extension FTN that start with the letters PROG.

PROG.- lists files named PROG with any extension.

-.-/ lists all system files and all files under the users account.

#### 4.3. CONTROL COMMANDS

##### 4.3.1. Commands to send a message to the system console:

TELL <message>

ASK <message>

These commands are not transactions and can be sent at any time. SERV32 prints the message on the operator console and, if ASK is specified, prompts for a reply. The reply message is forwarded to the user as a TELL message.

##### 4.3.2. Command to allow the user to cancel a transaction:

CANCEL

SERV32 permits the user to cancel the following transactions:

REPLACE

CREATE

APPEND

FETCH

The DELETE and RENAME transactions cannot be cancelled since the transactions are complete when accepted.

##### 4.3.3 Command to terminate the file transfer session:

LOGOUT

A LOGOUT will terminate the privileges granted at LOGIN time, but will not break the physical network connection. Once a LOGOUT command is received, SERV32 will not accept any transactions except a LOGIN.

#### 4.4 UNSUPPORTED TRANSACTIONS

1. SUBMIT
2. MAIL

#### 4.5 ERROR RESPONSES

SERV32 send two kinds of error responses.

1. A transaction reject when the error is encountered while processing a transaction request.
2. A cancel transaction when an I/O error occurs during a file transfer.

The cause of the error is forwarded to the user in the form of an FTP error code for computer program consumption, and explanatory text for the user. All assign, allocate, rename and I/O error status codes are included in reject or cancel explanatory text.

#### 4.6 FILE DESCRIPTORS

Perkin-Elmer file descriptors are composed of four fields:

volume, filename, extension, and class

Its syntax is:

voln:filename.ext/file class.

voln: is the name of a disk volume. It is from one to four characters long. The first character must be alphabetic, and the remaining ones must be alphanumeric.

Filename is the name of a file and is from one to eight characters long. The first character must be alphabetic, with the remaining ones being alphanumeric.

Ext is the name of the extension and is from one to three alphanumeric characters long.

File class is the class name of the file and is one character long. The following class names are supported:

P	private file
G	group file
S	system file

If file class is not specified, the default file class is P.

When the user expects SERV32 to create a file, the file descriptor can include all the arguments of the Perkin-Elmer allocate command:

```
voln:filename.ext/class,Contiguous,fsize,keys  
voln:filename.ext/class,INdex,lrecl/bsize/isize,keys
```

If INDEX is chosen, lrecl specifies the logical record length, bsize specifies the physical block size in 256-byte sectors to be used for buffering the file, and isize specifies the index block size in 256 bytes sectors.

If CONTIGUOUS is chosen, fsize specifies total allocation size in 256-byte sectors.

Keys are in the form of two hexadecimal bytes, the left byte is the write key and the right byte is the read key.

For a more detailed discussion of file descriptors, see the OS/32 MTM Reference Manual, S29-591

SERV32 honors the request even if no allocation fields are specified. Default allocation fields are INDEX, bsize = 1, isize = 1 and no keys. Record length (lrecl for INDEX) or file size (fsize for CONTIGUOUS) are requested from the user.

## 5. PERKIN-ELMER 16 NBSNET FILE TRANSFER SERVER

## SERV16

## 5.1. INTRODUCTION

SERV16 is a program that runs under the FLOXOS Operating System on Perkin-Elmer 7/16 computers and implements the NBSNET file transfer protocol. SERV16 never initiates a file transfer session, but waits for a remote user to initiate a session with it. The remote user requests the file transfer transactions, and SERV16 determines the validity of the transactions and executes valid transactions. Invalid transactions are rejected. A login transaction, which identifies the remote user, is required before any file transfer transactions are accepted. Only one transaction can be active at a time.

The following is a description of SERV16 implemented commands. These commands are sent to SERV16 from a FTP user program running on a distant computer. FILENAME represents the name of a FLOXOS file or a FLOXOS ALLOCATE or ACTIVATE specification, and comma is the field separator for all commands which have more than one argument. Refer to section 5.6 for details on the syntax of the file descriptors. The maximum record length supported by SERV16 is 256 characters. Other file parameters are used by some commands as default values. These include the physical device address, which is set to the current default physical device, and the character mode, which is set to ASCII. It is possible to change these defaults during a file transfer session. The description of each command will specify when these default parameters are used.

## 5.2. TRANSACTION COMMANDS

## 5.2.1. Transaction to request access to SERV16:

LOGIN site-id

SERV16 requires a site-id from the user. Site-id can be any name by which the user wishes to be known to the system and can consist of a maximum of 50 printable ASCII characters. If this parameter is not specified on the transaction command, SERV16 will query the user for it. A carriage return in response to this question causes the transaction to be rejected.

Upon receiving a user's site-id, SERV16 will send a signon line identifying itself. At this time the user can enter any valid transaction command.

## 5.2.2 Transaction which allows the user to specify a default physical device:

DEFAULT DEV=pp

SERV16 checks the syntax of the command, and if valid, checks to see if the physical device address (pp) specified is valid. If it is valid, the default



physical device is set to pp; otherwise the transaction is rejected. The DEFAULT DEV command without an argument will set the default physical device to the one set when SERV16 is first initiated. The default physical device is used for those transactions where no physical device address is specified.

### 5.2.3. Transaction to set I/O mode between the SERV16 and its file manager:

```
DEFAULT MOD=A
DEFAULT MOD=B
```

This command will set the character mode to either ASCII (MODE A) or binary (MODE B). The default mode is ASCII. In this mode, SERV16 requests ASCII I/O from its file manager. Upon receiving a MODE default, SERV16 sets the appropriate I/O mode. It first checks to be sure the syntax of the command and the argument is valid. The DEFAULT MOD command without an argument resets the mode to ASCII.

### 5.2.4. Transaction to determine the current defaults for SERV16:

```
DEFAULT
```

This transaction reports to the user the current defaults for the physical device and the mode of the I/O.

### 5.2.5. Transactions to transfer information from the user's file to the server's file:

```
REPLACE FILENAME
CREATE FILENAME
```

In the CREATE transaction, if FILENAME already exists, SERV16 rejects the transaction.

In either transaction, if FILENAME does not exist, SERV16 creates a new file. The user can include in FILENAME any or all of the FLOXOS ALLOCATE command fields. SERV16 will substitute the current default for the physical device if it is not specified, but will not default for record length or number of cylinders. If these are not specified, the user will be queried for them. A carriage return to any of these questions will cancel the transaction. Once the new file has been created, SERV16 accepts the transaction and writes incoming data into the new file.

### 5.2.6. Transactions to transfer information from the server's file or directory to the user's file:

```
FETCH FILENAME
DIRLIST pp
```

With the DIRLIST transaction, pp represents the physical device address whose directory is to be listed. The physical device address, if specified, must be valid. If not, the transaction is rejected. If no physical device address is specified, the current physical device is used. SERV16 creates a scratch file to which the directory is written. This file is allocated with

2 cylinders. SERV16 attempts to create a scratch file named \$DIRAA. If this file already exists, it will try \$DIRAB, \$DIRAC, etc. If unable to create the scratch file, SERV16 will reject the transaction. If the allocation is successful, the contents of the scratch file will be transferred to the user and, upon completion, the scratch file will be deleted.

For the FETCH transaction, if FILENAME exists, SERV16 accepts the transaction and transfers the contents of this file to the user. If FILENAME does not exist, the transaction is rejected.

5.2.7. Transaction to delete a file from the server's file system:

DELETE FILENAME

If FILENAME exists, it is deleted and the transaction is accepted. If FILENAME does not exist, the transaction is rejected. The format of FILENAME for this transaction may include all of the ACTIVATE command fields. Refer to section 5.6 for information on the ACTIVATE command. The default physical device will be used if not specified.

### 5.3. CONTROL COMMANDS

5.3.1. Commands to send a message to the system console:

TELL MESSAGE  
ASK MESSAGE

In both commands, SERV16 prints the message on the system console. If it is an ASK, SERV16 will also prompt for a reply from the console. The reply message is forwarded to the user by a TELL command. These commands are not transactions and can be received by SERV16 at any time.

5.3.2. Command to allow the user to cancel a transaction:

CANCEL

SERV16 permits the user to cancel the following transactions:

REPLACE  
CREATE  
FETCH  
DIRLIST

The DELETE transaction cannot be cancelled since the transaction is complete when accepted.

In the case of a REPLACE or CREATE, if a file has been created, the file will be deleted. With a REPLACE of an existing file, the data received before the cancel will be written to the file, but no filemark will be written.

### 5.3.3. Command to terminate the file transfer session:

END

An END command will terminate the current session, but will not break the physical network connection. Once an END command is received, SERV16 will not accept any transactions until a new connection is made from another remote computer. SERV16 will print a message to the console specifying the address of the remote computer which just disconnected.

## 5.4 UNSUPPORTED TRANSACTIONS

APPEND  
MAIL  
SUBMIT

## 5.5 ERROR RESPONSES

SERV16 sends two kinds of error responses:

1. A transaction reject when the error is encountered while processing a transaction request.
2. A cancel transaction when an I/O error occurs during a file transfer.

The cause of the error is forwarded to the user in the form of an FTP error code for use by a computer program, and explanatory text for the user. Assign and allocate error codes are included in reject or cancel explanatory text.

I/O error status codes are as follows:

X'CO'	a read or write operation was attempted on a file with a read or write protect attribute, or a write operation was attempted on a file with hardware write protect enabled.
X'A0'	an attempt was made to use a physical device which was unavailable.
X'90'	an end-of-medium was reached on a file.
X'84'	a parity error was detected on a read operation, or a defective track condition occurred.

## 5.6 FILE DESCRIPTORS

In the following descriptions, arguments in lower case are optional.

For those files that exist, the file descriptor specified can be of the same form as the FLOXOS ACTIVATE command:

NNNNNN,pp

where NNNNNN is the name of the disk file to be activated, and pp is the optional physical device address. The SERV16 default will be used if pp is not specified.

For those files that do not exist, the file descriptor specified can be of the same form as the ALLOCATE command:

NNNNNN,pp,ccc,1111

where NNNNNN is the name of the disk file to be allocated, pp is the optional physical device address (SERV16 default used if none is specified), ccc is number of cylinders to be allocated for the file and 1111 is the record length of the file in bytes (no default). The number of cylinders allocated will be 1 if this argument is not specified but the record length is specified.



## 6. SPERRY 1100 NBSNET FILE TRANSFER SERVER

## SERV1100

## 6.1 INTRODUCTION

SERV1100 is a real time program that runs on SPERRY 1100 series computers and implements the NBS file transfer protocol. This program does its own communications, account verification, accounting and scheduling. All of the overhead of the normal SPERRY interactive interface is bypassed. SERV1100 is a server program and as such never initiates a file transfer session, but waits for a remote user to initiate a file transfer session with it. The remote user requests file transfer transactions. The server determines the validity of the transaction and either executes valid ones, or rejects invalid ones. A valid LOGIN transaction is required before any file transfer transactions are accepted.

The SPERRY server can serve a limited number of users at a time. When the server receives a LOGIN transaction, it determines whether or not service facilities are available. If these facilities are not available, a LOGOUT is sent to the user and the network connection is broken. If service is available, the LOGIN transaction is processed. The server permits more than one transaction to be active, however only one file transfer in each direction can be active at one time.

When a user makes an FTP connection, he is interacting with the SPERRY 1100 currently running the SERV1100 program. If SERV1100 is not loaded and the user establishes an FTP connection, he will immediately receive a remote disconnect. The user should contact the SPERRY operator to have SERV1100 loaded.

SERV1100 will transfer standard data files or symbolic program file elements. A number of utility functions are also available. The user can delete files, submit a file containing a runstream for batch execution or request a listing of the table of contents of a program file. A default program file can be set permitting a single file assignment for multi-element transfers from the same program file.

Following is a list of transactions and other commands supported by SERV1100. FILENAME represents the name of a public SPERRY datafile or program file. Required arguments are capitalized. Abbreviations for the specification fields in the DEFAULT command are underlined>. A comma is the field separator for all commands which have more than one argument. A description of error messages is included and the syntax of SPERRY file descriptor, FILENAME, is explained. For a more detailed explanation of each transaction, see the FTP user manual written for the user's FTP program.

## 6.2 TRANSACTION COMMANDS

## 6.2.1. Transaction to request access to SERV1100:

```
LOGIN runid,account-number,project-id
```

LOGIN is equivalent to the run card in a regular batch or interactive SPERRY run. The user can include any of the fields on the LOGIN transaction. The server will prompt for any missing fields. If the account number is valid, the LOGIN transaction is accepted and SERV1100 sends a signon message identifying itself. If the user does not give a PROJECT on the LOGIN command or in answer to the server's request when the LOGIN is otherwise valid, the project 'QSQSQ\$' is assumed. The user now has read and write access to files on the SPERRY 1100 running SERV1100. If the account number is not valid, the LOGIN transaction is rejected. The network connection is not broken. The user can enter another LOGIN or disconnect.

## 6.2.2. Transaction to allow the user to specify a default program file.

```
DEFAULT DIRECTORY=filename.
```

If the requested file can be assigned and it is a program file or an empty file, the transaction is accepted. The user can now request file transfer of symbolic elements for this default program file by specifying element name only on the request.

## 6.2.3. Transactions which transfer information from the user's files to the server's files:

```
REPLACE FILENAME.element
CREATE FILENAME.element
```

With the REPLACE command, if FILENAME exists, the transaction is accepted. With the CREATE transaction, if FILENAME exists, the transaction is rejected.

For either transaction, if FILENAME does not exist, a new file is created under the account number specified in the LOGIN transaction.

When using the REPLACE command to transfer a symbolic element, FILENAME need not be specified if it was set as the default program file on the most recent DEFAULT transaction. A new element will be created in the requested program file, whether it exists or not. If the element did exist, the old element will be marked deleted.

## 6.2.4. Transaction to transfer information from the server's files to the user's files:

```
FETCH FILENAME.element
```

If a datafile is requested and FILENAME exists, the transaction is accepted and the contents of the file is sent to the user. If FILENAME does not exist, the transaction is rejected.

If a symbolic element is requested, the FILENAME need not be specified if it was set as the default program file on the most recent DEFAULT transaction. If the element exists, the transaction is accepted and the contents of the element are sent to the user's computer.

#### 6.2.5. Transaction to delete a file from the server's file system:

DELETE FILENAME.

If FILENAME exists, it is deleted and the transaction accepted.

#### 6.2.6. Transaction to start a batch job:

SUBMIT FILENAME.

If FILENAME exists and contains a valid runstream, the transaction is accepted and the run is submitted for later batch execution.

#### 6.2.7. Transaction to request a directory listing of a program file.

DIRLIST filename.

If the requested file exists and is a program file, the transaction is accepted and a table of contents of the requested program file is sent to the user. If filename is not specified and a default program file has been set by the DIR field specification on the DEFAULT command, the user receives the table of contents of the default program file.

### 6.3. CONTROL COMMANDS

#### 6.3.1. Commands to send a message to the system console:

TELL MESSAGE  
ASK MESSAGE

The TELL command will print the message on the operator console. It is a command that can be entered at any time, even before a LOGIN transaction.

The ASK command will print the message on the operator console and prompt for a reply. The reply is forwarded to the user as a TELL message. This command can only be entered after a successful LOGIN; otherwise it will be rejected.

#### 6.3.2. Command to allow the user to cancel a transaction:

CANCEL

SERV1100 permits the user to cancel the following transactions:

REPLACE  
CREATE  
FETCH  
DIRLIST



The DELETE and SUBMIT transactions cannot be cancelled because the transactions are complete when accepted.

### 6.3.3. Command to terminate the file transfer session:

LOGOUT

A LOGOUT will terminate the privileges granted at LOGIN time and release the service facilities to another user. The network connection is not broken. The user can enter another LOGIN or disconnect.

## 6.4 UNSUPPORTED TRANSACTIONS

APPEND  
MAIL

## 6.5 ERROR RESPONSES

SERV1100 sends two kinds of error responses.

1. A transaction reject when an error is encountered while processing a transaction request.
2. A cancel transaction when an I/O error occurs during a file transfer.

The cause of the error is forwarded to the user in the form of an FTP error code for computer program consumption and explanatory text for the user. For instance, if a file cannot be assigned, the appropriate message from the following list is sent with the reject transaction.

NO READ PERMISSION  
NO WRITE PERMISSION  
ILLEGAL FILENAME  
FILE IS NOT CATALOGUED  
FILE IS ROLLED OUT  
FILE ASSIGNED EXCLUSIVELY TO ANOTHER RUN  
KEYS CONFLICT  
PRIVATE FILE  
FILE WAS PREVIOUSLY CATALOGUED  
OTHER FAC REJECT

The I/O error code is included in the explanatory text of the cancel messages.

## 6.6 FILE AND ELEMENT DESCRIPTORS

The following notation is used to specify SPERRY file names:

qualifier\*filename(F-cycle)/read-key/write-key

The qualifier and the filename fields each can be up to twelve alphanumeric characters. The special characters '-' and '\$' are also acceptable

characters. F-cycle is an integer number. Read-key and write-key can be any character in the Fieldata character set except blank, comma, slash, period, and semicolon.

The following notation is used to specify SPERRY element names:

element/version

The element and version names can each be up to twelve alphanumeric characters. The special characters '-' and '\$' are also acceptable characters. Note: element cycle specifications are not recognized by SERV1100.

For a more detailed discussion of SPERRY file descriptors refer to the SPERRY Programmer Reference Manual UP-4144.2.





U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> <i>(See instructions)</i>	<b>1. PUBLICATION OR REPORT NO.</b> NBSIR 84-2970.	<b>2. Performing Organ. Report No.</b>	<b>3. Publication Date</b> November 1984
<b>4. TITLE AND SUBTITLE</b> <p style="text-align: center;">File Transfer Over the NBSNET User's Manual</p>			
<b>5. AUTHOR(S)</b> <p style="text-align: center;">Debra Strawbridge and Carol Young</p>			
<b>6. PERFORMING ORGANIZATION</b> <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		<b>7. Contract/Grant No.</b>	<b>8. Type of Report &amp; Period Covered</b>
<b>9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS</b> <i>(Street, City, State, ZIP)</i>			
<b>10. SUPPLEMENTARY NOTES</b>  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
<b>11. ABSTRACT</b> <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> <p>File Transfer Protocol (FTP) is a set of specifications for the transfer of information between computers via the NBS local area network. A description is given for the implementations of the File Transfer Protocol for Perkin Elmer*32-bit and 16-bit computers and for the UNIVAC 1100 series computers. User instruction and reference documentation are given for both the file transfer initiating programs and the remote host-interface programs. Sample file transfer sessions, description of error messages and summary of commands are included.</p> <p>* Certain trade names and company products are mentioned in the text. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the products are necessarily the best available for the purpose.</p>			
<b>12. KEY WORDS</b> <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> File Transfer, FTP, Network, Data Communications, Minicomputers, Server, File Transfer Protocol			
<b>13. AVAILABILITY</b> <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		<b>14. NO. OF PRINTED PAGES</b> 44	<b>15. Price</b> \$8.50





