FTP-RSX
Software User's Guide

Version 2.2-5

MAN/FTPRSXUG–94

First Printing, March 1985
Revised Printing, May 1989
Revised Printing, July 1990
Revised Printing, July 1992
Revised Printing, May 1993

Contents

1 Introduction

1.1 What Is FTP-RSX? 1
1.1.1 Client and Server Support 1
1.1.2 Protocols 1
1.2 TCPware) for RSX Family Of Software Products 1
1.3 Connecting Dissimilar Systems 2
1.4 Intended Audience 4
1.5 Prerequisite Knowledge 4
1.6 Contents 4
1.7 The Documentation Set 5
1.8 Conventions 5
1.9 Software License 5
1.10 Software Registration Card 5
1.11 Software Update Service 6

2 Installation

3 Transferring Files

3.1 Preparing to Transfer Files 9
3.2 Running UFTP 10
3.3 Opening a Connection 10
3.4 UFTP GET and PUT Commands 10
3.5 Exiting UFTP 12
3.6 Additional UFTP Commands 12
3.7 Sample UFTP Session 13

4 General Information

4.1 Network Hardware Configuration 15
4.2 Communications Protocol 15
4.3 Component Overview 15
4.3.1 User-FTP 16
4.3.2 User-Written Applications 16
4.3.3 Server-FTP 17
4.3.4 FTP-RSX Library Routines 17
4.3.5 IN Device Driver, INSACP, and NETCU 17
4.3.6 Network Device Driver 17
4.4 References 18

5 User-FTP Utility

5.1 Starting User-FTP . . . 19
5.2 UFTP Commands . . . 19
5.2.1 ASCII Command . . . 20
5.2.2 BELL Command . . . 20
5.2.3 BINARY Command . . . 20
5.2.4 CD Command . . . 20
5.2.5 CLOSE Command . . . 21
5.2.6 DEBUG Command . . . 21
5.2.7 DELETE Command . . . 21
5.2.8 DIR Command . . . 22
5.2.9 GET Command . . . 22
5.2.10 HELP Command . . . 23
5.2.11 LS Command . . . 23
5.2.12 MDELETE Command . . . 24
5.2.13 MGET Command . . . 24
5.2.14 MKDIR Command . . . 25
5.2.15 MPUT Command . . . 25
5.2.16 OPEN Command . . . 26
5.2.17 PUT Command . . . 26
5.2.18 PROMPT Command . . . 27
5.2.19 PWD Command . . . 27
5.2.20 QUIT Command . . . 28
5.2.21 QUOTE Command . . . 28
5.2.22 REMOTEHELP Command . . . 29
5.2.23 RENAME Command . . . 29
5.2.24 RMDIR Command . . . 29
5.2.25 STATUS Command . . . 30
5.2.26 TYPE Command . . . 30
5.2.27 USER Command . . . 32
5.3 UFTP Messages . . . 32
5.4 Aborting Commands and File Transfers . . . 33
5.5 Using Indirect Command Files . . . 33
5.6 Library Routine Usage . . . 34

6 FTP Library Reference

6.1 Library Routines . . . 36
6.1.1 FTPABO Routine . . . 36
6.1.2 FTPCCL Routine . . . 36
6.1.3 FTPCWD Routine . . . 37
6.1.4 FTPDEL Routine . . . 37
6.1.5 FTPDIR Routine . . . 37
6.1.6 FTPDNL Routine . . . 38
6.1.7 FTPGET Routine . . . 39
6.1.8 FTPINI Routine . . . 39
6.1.9 FTPLIN Routine . . . 40
6.1.10 FTPOOC Routine . . . 40
6.1.11 FTPPUT Routine . . . 41

6.1.12 FTPQUO Routine . . . 41
6.1.13 FTPSFT Routine . . . 42
6.2 Status Codes . . . 42
6.3 Linking with the Library . . . 43
6.4 Debugging Support . . . 43

7 Server-FTP (FTPD)

7.1 FTP Commands . . . 45
7.2 Site-Specific Command . . . 46
7.3 Event Logging Subroutine . . . 47

A Sample Program

B Implementation Notes

B.1 File Transfer Protocol (FTP) . . . 51
B.2 Memory Usage . . . 53

C Release Notes

C.1 Version 2.2-5 . . . 55
C.2 Version 2.2-4 . . . 55
C.3 Version 2.2 . . . 56
C.3.1 Commands . . . 56
C.3.2 Library Routines . . . 56
C.3.3 Additional Improvements . . . 56
C.4 Version 2.1 . . . 57
C.5 Version 2.0 . . . 57

Index

1 Introduction

1.1 What Is FTP-RSX?

FTP-RSX is a software product that provides TCP/IP File Transfer Protocol networking services for DEC PDP-11 computers running the RSX-11M, RSX-11M-PLUS, and MicroRSX operating systems.

1.1.1 Client and Server Support

FTP-RSX supports both the client (user) and server ends of the File Transfer Protocol. Files can be transferred in both directions between local and remote hosts that implement TCP/IP.

1.1.2 Protocols

FTP-RSX implements the following protocols:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Internet Protocol (IP)
- File Transfer Protocol (FTP)

The Department of Defense (DoD) has adopted these protocols as standards for use in all packet networks.

1.2 TCPware® for RSX Family Of Software Products

FTP-RSX is part of the TCPware for RSX family of software products. This family also includes TCP-RSX and TELNET-RSX, described in the following table.

Product	Purpose
TCP-RSX	Provides the lower level network protocols (TCP and IP), support utilities, and a QIO programming interface.
TELNET-RSX	Allows local users to log in to remote hosts, and remote users to log in to hosts running TELNET-RSX. Supports both the client (user) and server ends of the TELNET protocol.

TCP-RSX is included with the FTP-RSX and TELNET-RSX software products.

1.3 Connecting Dissimilar Systems

FTP-RSX is ideal for connecting RSX computers to computer systems from many different manufacturers because the TCP/IP and FTP protocols are widely implemented, as shown in Figure 1-1. Almost all UNIX-based systems support TCP/IP and FTP.

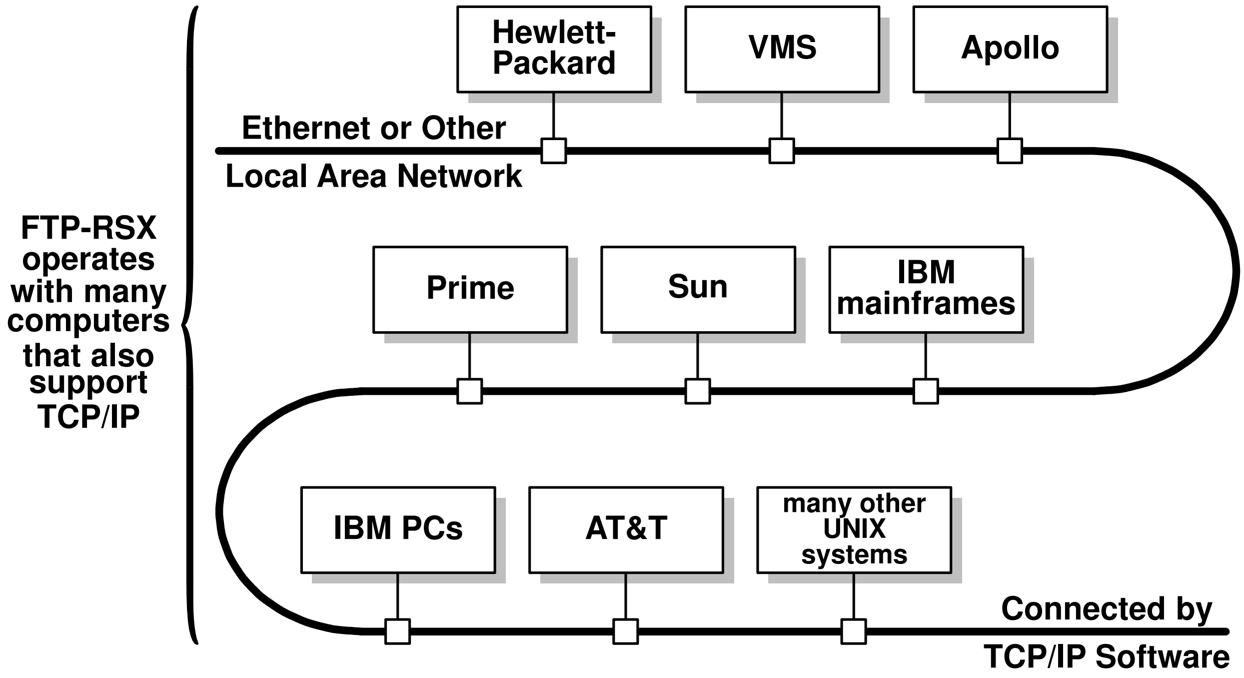

Figure 1-1: Computers That Support TCP/IP

FTP-RSX also operates with the many network support devices that are compatible with TCP/IP, Ethernet, and other Local Area Networks, as shown in Figure 1-2.

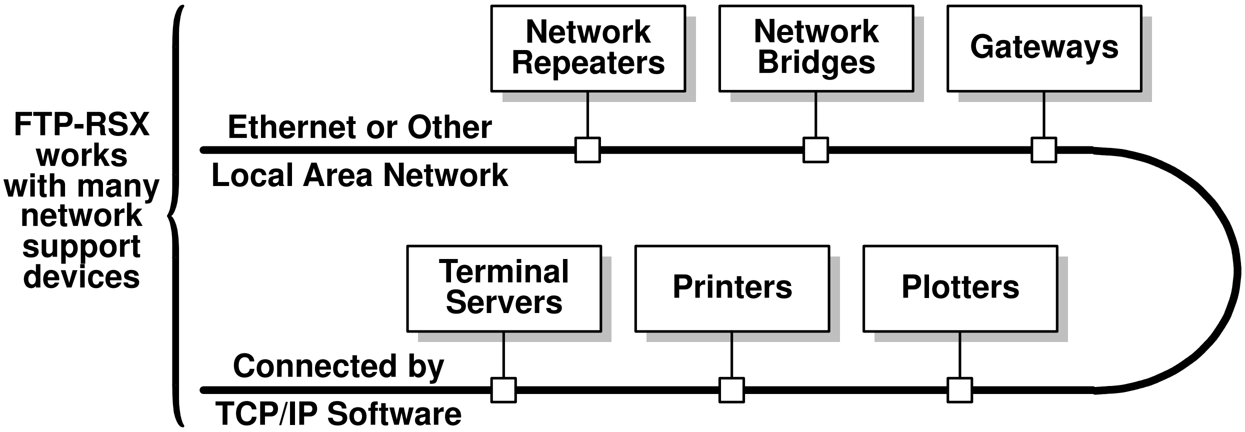

Figure 1-2: Devices that Support TCP/IP

Process Software Corporation develops TCPware for operating systems made by Digital Equipment Corporation (DEC), as shown in Figure 1-3.

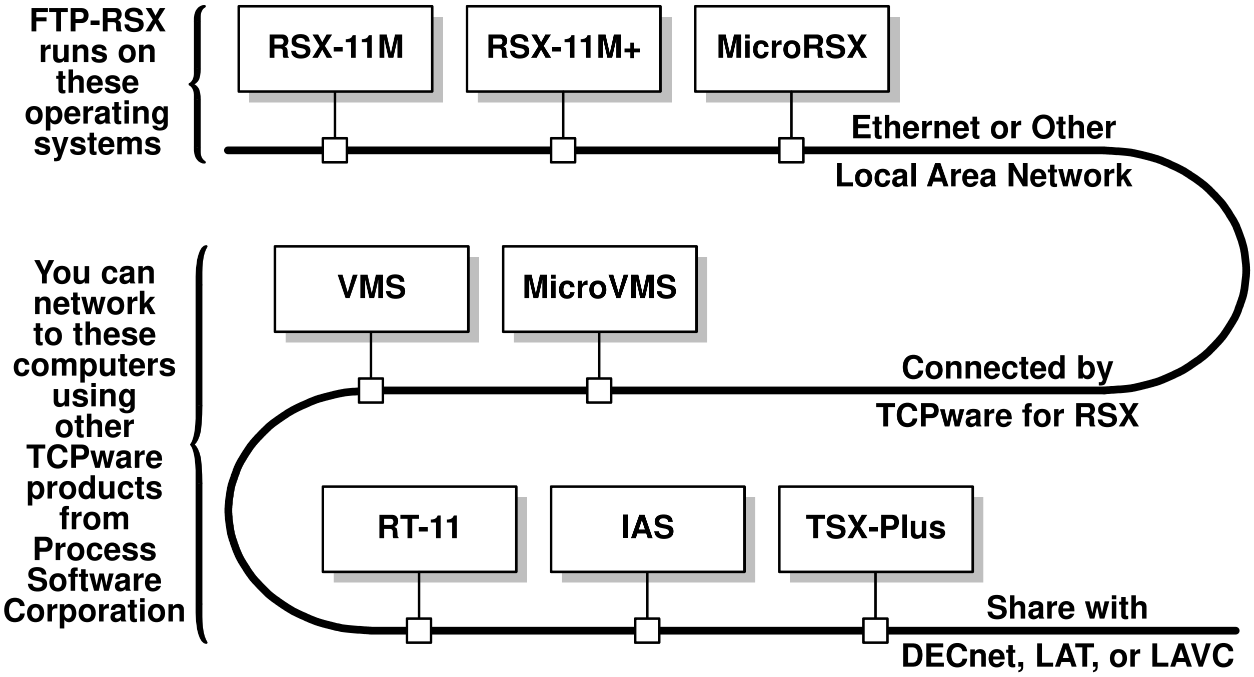

Figure 1-3: TCPware Connecting Different DEC Operating Systems

1.4 Intended Audience

The *FTP-RSX Software User's Guide* is for:

- system managers who install and maintain FTP-RSX
- application programmers who use the FTP-RSX library routines
- users who need to transfer files between local and remote computers

1.5 Prerequisite Knowledge

FTP-RSX users should be familiar with:

- the DEC RSX-11M, RSX-11M-PLUS, or MicroRSX operating systems
- the file systems for these operating systems

Some familiarity with computer networks is helpful, but not required.

This manual does not explain the operation of RSX-11M, RSX-11M-PLUS, or MicroRSX.

1.6 Contents

The *FTP-RSX Software User's Guide* contains the following chapters and appendices:

- *Introduction* provides a short overview of FTP-RSX and TCP/IP networking.
- *Installation,* for system managers, summarizes the steps required to install FTP-RSX. Complete installation instructions are in the *TCPware for RSX Software Reference Manual.*
- *Transferring Files* explains how to transfer files between computer systems using the User-FTP utility.
- *General Information* describes the network hardware and software configuration. It summarizes the purpose of User-FTP, Server-FTP, FTP-RSX library routines, IN device driver, INSACP, and the NETCU utility.
- *User-FTP Utility* explains what the FTP-RSX client does and how to use it.
- *FTP Library Reference*, for application programmers, describes the FTP-RSX library routines.
- *Server-FTP (FTPD)* explains what the FTP-RSX server does and how to use it.
- *Appendix A, Sample Program* provides a sample program to show how to use FTP-RSX library routines.
- *Appendix B, Implementation Notes* describes how FTP-RSX implements the FTP protocol.
- *Appendix C, Release Notes* summarizes the enhancements made to each version of FTP-RSX.

1.7 The Documentation Set

The complete documentation set for the TCPware for RSX family of software products consists of the following manuals:

- *TCPware for RSX Software Reference Manual*, which explains how to install and configure TCPware, use the support utilities, and use the QIO programming interface.
- *FTP-RSX Software User's Guide*, which explains how to use the FTP-RSX software product to transfer files between systems.
- *TELNET-RSX Software User's Guide*, which explains how to use the TELNET-RSX software product to access remote systems.

1.8 Conventions

This manual uses the conventions listed below.

`monospaced type`	Monospaced type is used in examples of system output or user input.
UPPERCASE	Uppercase letters are used in commands and examples to indicate that you should enter the characters shown. The system accepts uppercase or lowercase characters.
italics	Italics are used in commands and examples to indicate that either the system supplies or you should supply a value.
[ ]	Square brackets indicate the enclosed item is optional. Do not type the brackets when you enter the command.
^	A caret denotes a control character (ASCII characters from 1 to 31). To type a control character, press the "Ctrl" key and type the character that follows the caret.
Host	"Host" refers to any computer on the network. The local host is the computer you are currently using (your RSX system). A remote host is any other computer on the network.

1.9 Software License

Each copy of FTP-RSX includes a *Software License Agreement*. You should read and understand the *Agreement* before installing FTP-RSX.

The *Agreement* entitles you to install FTP-RSX on one computer system. If you want to use FTP-RSX on more than one computer, you must purchase additional FTP-RSX software products or licenses. Contact Process Software Corporation or your distributor for more information.

1.10 Software Registration Card

A Software Registration Card is enclosed with the FTP-RSX software product kit. You should fill out and return this card to Process Software Corporation to receive warranty support and software update information.

1.11 Software Update Service

Process Software Corporation periodically updates and enhances TCPware for RSX products to incorporate new features, fix bugs, and improve the documentation. The Software Update Service ensures that you receive the latest versions as they are released. FTP-RSX and TELNET-RSX each include updates for TCP-RSX. Contact Process Software Corporation to order this service.

2
Installation

This chapter is for system managers. It summarizes the steps required to install the FTP-RSX software product. Complete installation instructions are in the *TCPware for RSX Software Reference Manual.*

To install FTP-RSX, perform these steps:

1. **Copy** the distribution kit to a directory on the system.
2. **Build** the components for the product.
3. **Configure** the software.
4. **Obtain** a software password.
5. **Start** TCPware.
6. **Test** TCPware.
7. **Clean up** after the installation.

3
Transferring Files

This chapter describes how to transfer files between your local RSX system and a remote host.

The *Sample UFTP Session* section in this chapter shows you what a UFTP session is like.

3.1 Preparing to Transfer Files

Before you transfer files, you will need to know:

- The name of the remote host that you want to transfer files to or from. The remote host must exist on the network and have an FTP server running.

  One way to find out this information is to use the Network Control Utility to display the Host Name Table.

  Follow these steps to display the list of hosts on the network:

  1. Run the Network Control Utility by typing:

     `NCU`

  2. At the “NETCU>” prompt, type:

     `LIST`

     This displays a list of the host names on your network. You need not be concerned with the internet addresses.

  3. Exit the Network Control Utility by typing:

     `EXIT`

     The Network Control Utility is described in the *TCPware for RSX Software Reference Manual.*

- The username and password for an account on the remote host. If the remote host does not support multi-user protection features, you may not need a username and password.
- The conventions for naming files on the remote host.

3.2 Running UFTP

To run the User-FTP utility (UFTP), type the following at the DCL/MCR prompt:

```
UFTP
```

The UFTP prompt “UFTP>” should display. Proceed to the next section to open a connection to a remote host.

If the “UFTP>” prompt does not display, ask your system manager whether FTP-RSX was installed and started as described in the *TCPware for RSX Software Reference Manual*.

3.3 Opening a Connection

Before transferring files, you must open a connection to the remote host by using the UFTP OPEN command.

The format for the OPEN command is:

OPEN *host-name*

where *host-name* is the name of the remote host you want to connect to.

You are prompted for your username and password on the remote host. The password will not be displayed as you type it. If the remote host does not support multi-user protection features, any username and password you type is acceptable.

The connection remains open until you exit UFTP, close the connection with the CLOSE command, or try to open a new connection.

For example, suppose you want to connect to the remote host SYS1. Your username is “SMITH”, and your password is “SECRET”. The following command opens the connection.

OPEN SYS1

When prompted for your username, type `SMITH`. When prompted for your password, type `SECRET`.

Once the connection is open, you can proceed to the next section to transfer files using the GET and PUT commands.

3.4 UFTP GET and PUT Commands

Use the UFTP GET and PUT commands to transfer files. The GET command is used to transfer a file from the remote host to the local host. The PUT command is used to transfer a file from the local host to the remote host.

Figure 3-1 shows the transfer direction for the GET and PUT commands:

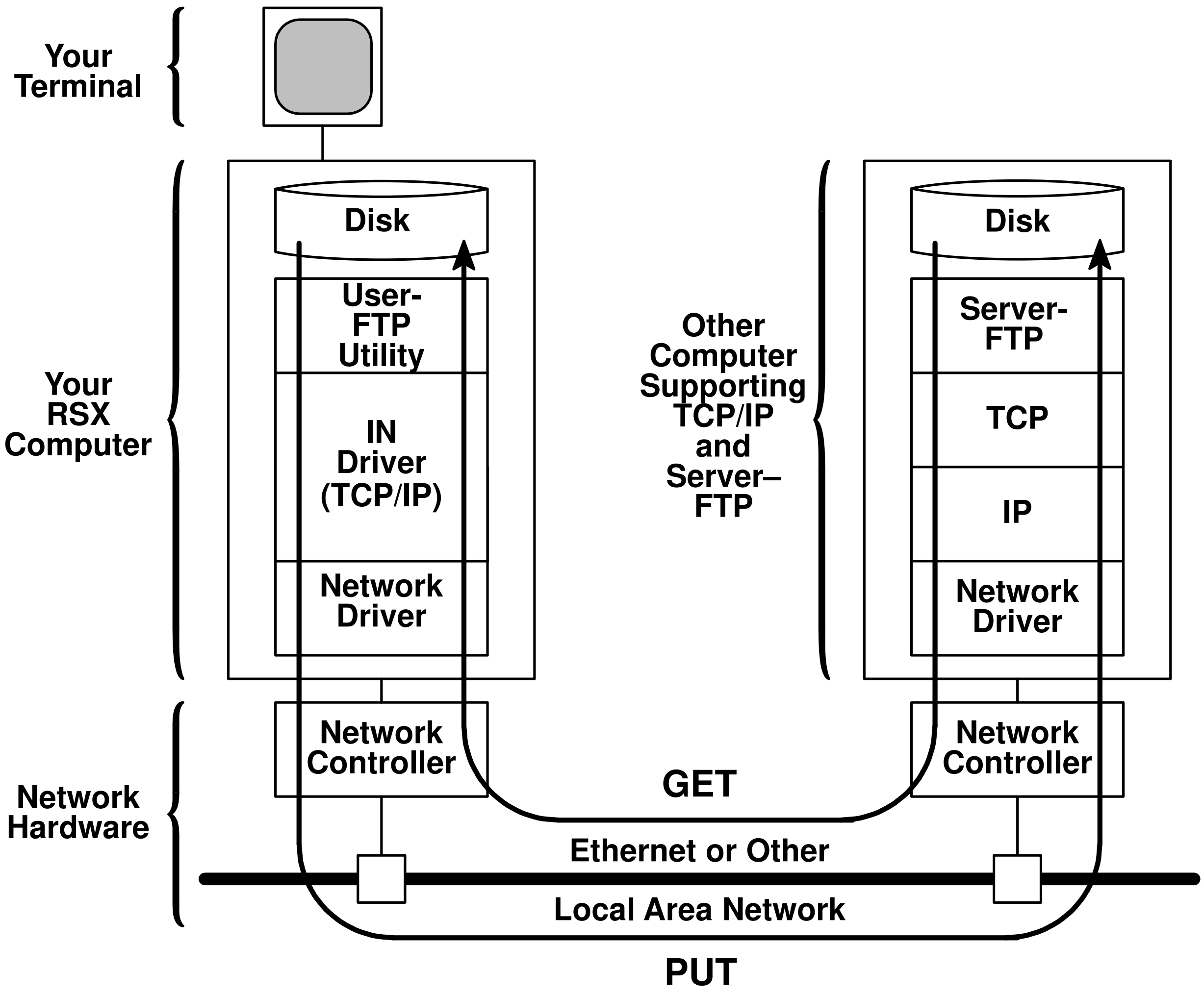

Figure 3-1: GET and PUT Transfer Direction

The format for the GET command is:

GET *remote-file* [*local-file*]

The format for the PUT command is:

PUT *local-file* [*remote-file*]

where *local-file* is a file specification on your local host and *remote-file* is a file specification on the remote host.

Remote file specifications may have to be enclosed within double quotes if they contain special symbols.

If the optional file specification is omitted, it is taken from the first file specification.

For example, suppose you want to copy the local file STUFF.TXT to the remote host. To transfer the file, type:

```
PUT STUFF.TXT
```

The remote file is created in the current working directory for the account you logged into when the connection was opened. The remote file has the same file name and extension as the source (STUFF.TXT) because you did not specify a remote file name and extension.

The following command transfers the file sample.pas from the directory /usr/src/ on a remote UNIX system to the local file SAMPLE.PAS:

```
GET "/usr/src/sample.pas" SAMPLE.PAS
```

For complete information on the UFTP GET and PUT commands, refer to the *User-FTP Utility* chapter later in this document.

NOTE

The second (optional) file specification for GET and PUT commands must be complete if specified. The filename and file extension from the first file specification are *not* used as defaults for the second file specification. Therefore, the command GET FILE.DAT [100,30] would create the file [100,30].TXT *not* [100,30]FILE.DAT.

3.5 Exiting UFTP

Exit UFTP by typing:

```
QUIT
```

This returns you to the DCL/MCR prompt. If a connection is open, it will be closed before exiting. To transfer files once you exit UFTP, you must run UFTP again.

3.6 Additional UFTP Commands

The following are additional commands that you can use while in the UFTP utility. The *User-FTP Utility* chapter describes all of the UFTP commands in detail.

- **ASCII** - Change the file transfer representation type to ASCII.
- **BELL** - Toggles a bell sound after a file transfer.
- **BINARY** - Change the file transfer representation type to binary.
- **CD** - Change your remote working directory.
- **CLOSE** - Closes an open connection.
- **DELETE** - Delete a file on the remote host.
- **DIR** - Display a remote directory.
- **GET** - Transfers a file from the remote host to the local host.
- **HELP** - Display the UFTP help text.
- **LS** - Lists the files in a remote directory.

- **MDELETE** - Deletes multiple files from a remote host.
- **MGET** - Transfers multiple files from the remote host to the local host.
- **MDIR** - Creates a directory on the remote host.
- **MPUT** - Transfers multiple files from the local host to the remote host.
- **OPEN** - Opens a connection to a remote host.
- **PUT** - Transfers a file from the local host to the remote host.
- **PROMPT** - Toggles prompting for the MGET, MPUT, and MDELETE commands.
- **PWD** - Displays the name of the current remote directory.
- **QUIT** - Exits from the UFTP utility and returns you to the DCL/MCR prompt.
- **QUOTE** - Send an arbitrary command to the remote Server-FTP.
- **REMOTEHELP** - Displays help on the remote server.
- **RENAME** - Renames a file on the remote host.
- **RMDIR** - Deletes a directory on the remote host.
- **STATUS** - Display information about your current UFTP session.
- **TYPE** - Change the file transfer representation type.
- **USER** - Send new login information to the remote host.

3.7 Sample UFTP Session

The following sample terminal session uses UFTP to:

1. Open a connection to remote host ALPHA.
2. Transfer files.
3. Delete a remote file.
4. Demonstrate additional UFTP commands.

The terminal session is performed on a RSX-11M-PLUS system. The remote host is a DEC ULTRIX system.

Text you type is underlined.

```
> UFTP
UFTP   V2.2  Copyright (c) 1990 Process Software Corporation
UFTP> OPEN ALPHA
220 alpha FTP server (Ultrix Version 4.10 Mon Nov 7 15:52:11 EST 1988) ready.
Username ? smith
```

```
Password ?
331 Password required for smith.
230 User smith logged in.
UFTP> DIR
200 PORT command successful.
150 Opening data connection for /bin/ls (223.42.95.70,1046) (0 bytes).

total 5
-rwxr-xr-x  1 smith    users         281 May  8 19:43 .cshrc
-rwxr-xr-x  1 smith    users         340 May  8 19:43 .login
-rwxr-xr-x  1 smith    users         138 May  8 19:43 .profile
drwxr-xr-x  2 smith    users         512 May  8 19:43 bin
-rw-r--r--  1 smith    users         392 May 29 11:20 file1
-rw-r--r--  1 smith    users         400 May 29 11:30 file3

226 Transfer complete.
UFTP> GET file1
200 PORT command successful.
150 Opening data connection for file1 (223.42.95.70,1047) (392 bytes).
226 Transfer complete.
UFTP> PUT file2
200 PORT command successful.
150 Opening data connection for file2 (223.42.95.70,1048).
226 Transfer complete.
UFTP> DELETE file3
250 DELE command successful.
UFTP> DIR
200 PORT command successful.
150 Opening data connection for /bin/ls (223.42.95.70,1049) (0 bytes).

total 5
-rwxr-xr-x  1 smith    users         281 May  8 19:43 .cshrc
-rwxr-xr-x  1 smith    users         340 May  8 19:43 .login
-rwxr-xr-x  1 smith    users         138 May  8 19:43 .profile
drwxr-xr-x  2 smith    users         512 May  8 19:43 bin
-rw-r--r--  1 smith    users         392 May 29 11:20 file1
-rw-r--r--  1 smith    users          22 May 29 12:55 file2

226 Transfer complete.
UFTP> QUOTE "PWD"
257 "/usr/users/smith" is current directory.
UFTP> CLOSE
221 Goodbye.
UFTP> QUIT
> DIR

Directory DU0:[SMITH]
29-MAY-90 12:56

FILE1.;1           1.         29-MAY-90 09:37
FILE2.;1           1.         22-MAY-90 09:38

Total of 2./2. blocks in 2. files

>
```

4
General Information

This chapter provides background information on FTP-RSX, including an overview of the major software components and how they interact.

4.1 Network Hardware Configuration

To operate the software, each computer should be configured with at least one of the supported network controllers and associated interconnecting hardware and cabling. The hardware components are essentially identical on each computer system.

During the file transfer operations, the communications network is configured logically as a two-node, point-to-point link between the two computers involved in the communication. The network may be physically part of a larger multi-node network with no impact on the other nodes in the network.

A file transfer operation is initiated by an operator or command procedure at only one of the computers. During file transfer, the other computer services the request of the initiating computer.

The *TCPware for RSX Software Reference Manual* further describes the supported network controllers.

4.2 Communications Protocol

FTP-RSX uses the Transmission Control Protocol/Internet Protocol (TCP/IP) and File Transfer Protocol (FTP) to effect file transfers. Use of these protocols provides a reliable and efficient means for moving files between computer systems.

The Department of Defense (DoD) has adopted the IP and TCP protocols as DoD-wide standards for all DoD packet networks. All new DoD packet networks will be using these protocols exclusively.

4.3 Component Overview

The FTP-RSX software is organized into a number of components. Figure 4-1 shows how they interrelate.

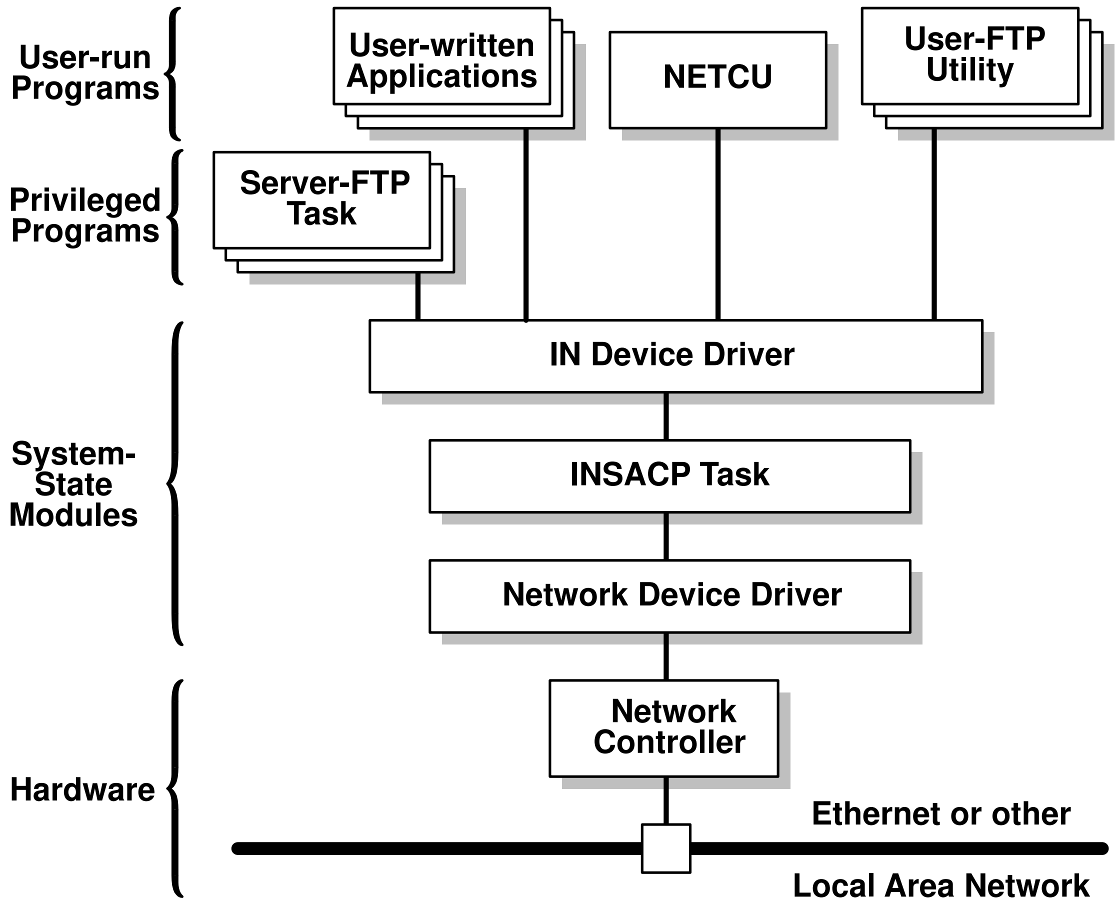

Figure 4-1: FTP-RSX Components

The purpose and function of each component are described below.

4.3.1 User-FTP

The User-FTP utility (UFTP) is the user's interface to FTP-RSX. The utility accepts commands that transfer files between computer systems and controls the method by which the files are transferred.

You can run UFTP directly (interactively) or indirectly from a command procedure. Any number of people can use UFTP, if memory is available for the connections.

The User-FTP utility is described in the *Transferring Files* and *User-FTP Utility* chapters.

4.3.2 User-Written Applications

The user-written applications can call FTP-RSX library routines or use TCP-RSX programming services. Refer to the *FTP Library Reference* chapter in this document and *TCP Services* chapter of the *TCPware for RSX Software Reference Manual* for more information.

4.3.3 Server-FTP

The Server-FTP task, called FTPD, services file transfer requests from a remote host.

The Server-FTP runs unattended. It "listens" for a connection from a remote User-FTP and then establishes communication between the local and remote hosts. More than one Server-FTP can be active at a time.

The Server-FTP restricts file access from remote hosts by using the account name and password supplied to it by a user on the remote host. The account name and password determine which files, accounts, and devices can be accessed according to the privileges of that account. This protects files on the local host from being accessed by unauthorized users.

The FTP protocol commands supported by the server are described in the *Server-FTP (FTPD)* chapter.

4.3.4 FTP-RSX Library Routines

The FTP-RSX library routines are a set of routines that you can link with your programs to perform FTP operations.

The *FTP Library Reference* chapter provides more information on using the FTP-RSX library routines.

The User-FTP utility uses the FTP-RSX library routines.

4.3.5 IN Device Driver, INSACP, and NETCU

The IN device driver, INSACP, and NETCU all work together to provide the lower protocol layers of the TCP/IP protocol stack (TCP, UDP, IP, ARP).

The IN device driver implements most of the lower layer protocols. INSACP provides an interface to the network device for the IN device driver and provides some support functions for IN. The NETCU utility is used to start, stop, and control the networking software.

The IN device driver provides the user/TCP and user/UDP programming services and is used by the User-FTP utility, Server-FTP, and your application programs to communicate with other systems on the network using the TCP/IP protocols.

The user/TCP and user/UDP programming services are described in the *IN Device Driver Services* chapter of the *TCPware for RSX Software Reference Manual*. The NETCU utility commands are described in the *Network Control Utility* chapter of the *TCPware for RSX Software Reference Manual*.

4.3.6 Network Device Driver

The network device driver operates the network controller hardware.

The *TCPware for RSX Software Reference Manual* contains more information on the supported network devices.

4.4 References

For more information on the Network Control Utility, the IN device driver, and how to use the IN device driver for task-to-task communications, refer to the *TCPware for RSX Software Reference Manual.*

For information on TELNET-RSX, refer to the *TELNET-RSX Software User's Guide.*

More information on the RSX-11M, RSX-11M-PLUS, and MicroRSX operating systems and utilities can be found in the appropriate DEC documentation.

More information on the FTP protocol can be found in the Request for Comments (RFC) documents published by the Network Information Center (NIC) at Government Systems, Inc., 14200 Park Meadow Drive, Suite 200, Chantilly, VA 22021 (800-365-3642, 703-802-4535 (7:00 AM to 7:00 PM EST)). You can also send electronic mail directed to the "HOSTMASTER" and "REGISTRAR" mailboxes at network address NIC.DDN.MIL. The RFCs contain the specifications for all internet protocols.

FTP-RSX implements the specification described in *File Transfer Protocol*, RFC-959.

5
User-FTP Utility

This chapter describes how to operate the User-FTP utility, UFTP.

You use UFTP to transfer ASCII and binary files to or from a remote host.

5.1 Starting User-FTP

To start User-FTP, type the following at the DCL/MCR prompt:

```
UFTP
```

The “UFTP>” prompt displays. This means UFTP is ready to accept your commands.

5.2 UFTP Commands

You interact with UFTP by typing the following commands at the “UFTP>” prompt:

- **ASCII**
- **BELL**
- **BINARY**
- **CD**
- **CLOSE**
- **DEBUG**
- **DELETE**
- **DIR**
- **GET**
- **HELP**
- **LS**
- **MDELETE**
- **MGET**
- **MKDIR**
- **MPUT**
- **OPEN**
- **PUT**
- **PROMPT**
- **PWD**
- **QUIT**
- **QUOTE**
- **REMOTEHELP**
- **RENAME**
- **RMDIR**
- **STATUS**
- **TYPE**
- **USER**

Command parameters (such as remote filenames or directories) that contain special characters may have to be enclosed in double quotes.

UFTP prompts for required command parameters if you do not specify them, but not for optional command parameters if you omit them.

Each UFTP command and its parameters are described in the following sections.

5.2.1 ASCII Command

Use the ASCII command to set the file transfer representation type to ASCII. See the TYPE command for more information. This is the default type.

Format

ASCII

5.2.2 BELL Command

Use the BELL command to toggle bell mode. When enabled, a bell sounds after each transfer command is completed. The initial setting is off.

Format

BELL

5.2.3 BINARY Command

Use the BINARY command to set the file transfer representation type to binary. See the TYPE command for more information. The default type is ASCII.

Format

BINARY

5.2.4 CD Command

Use the CD command to change the current remote working directory.

Format

CD [*remote-directory*]

Parameter

remote-directory

A directory specification on the remote host. Your current remote working directory is changed to *remote-directory*.

If you omit *remote-directory*, your current remote working directory is changed to the parent directory.

5.2.5 CLOSE Command

Use the CLOSE command to close an open connection.

Format

CLOSE

If a connection is not open, this command has no effect.

You can use DISCONNECT as a synonym for CLOSE.

5.2.6 DEBUG Command

Use the DEBUG command to enable or disable debugging mode.

Format

DEBUG *mask*

Parameter

mask

Specifies the bits to enable or disable display of received and transmitted FTP commands between the UFTP utility and the remote server. See the Debugging Support section of the *FTP Library Reference* chapter.

Examples

The following command disables display of received and transmitted commands:

```
DEBUG 0
```

The following command enables display of received replies:

```
DEBUG 20
```

The following command enables display of transmitted commands:

```
DEBUG 40
```

The following command enables display of received and transmitted commands:

```
DEBUG 60
```

5.2.7 DELETE Command

Use the DELETE command to delete a remote file.

Format

DELETE *rfilename*

Parameter

rfilename

A remote file specification.

Examples

The following command deletes the file STUFF.TXT from the current remote directory:

```
DELETE STUFF.TXT
```

The following command deletes the file STUFF.TXT from the remote directory [SMITH.MEMOS]:

```
DELETE [SMITH.MEMOS]STUFF.TXT
```

5.2.8 DIR Command

Use the DIR command to list the contents of a remote directory. The listing typically includes the file name, size, and date and time of creation for each file.

Format

DIR [*remote-directory* [*lfilename*]]

Parameters

remote-directory

A remote directory specification.

If you omit *remote-directory*, the current remote working directory is listed.

lfilename

A local file specification. This file receives the directory listing from the remote host.

If you omit *lfilename*, the remote directory listing displays on your terminal.

Examples

The following command displays on your terminal a list of files in the current working directory of the remote host:

```
DIR
```

The following command transfers a list of filenames from the current working directory of the remote host to the local file DIR.TXT:

```
DIR "" DIR.TXT
```

You must use two double quotes to specify a null parameter.

5.2.9 GET Command

Use the GET command to transfer a file from a remote host to your local host.

Format

GET *rfilename* [*lfilename*]

You can use RECV as a synonym for GET.

See the TYPE command for information on file transfer representations.

Parameters

rfilename

A remote file specification. This identifies the remote file that is being transferred to the local host.

lfilename

A local file specification. This is the name of the local file that is receiving the remote file.

If you omit *lfilename*, the remote file specification (*rfilename*) is used as the local file specification.

You must specify *lfilename* if the remote file specification does not conform to the local file-naming conventions.

Example

The following example transfers the remote file RECORDS.DAT to the local file RECRDS.DAT:

```
GET RECORDS.DAT RECRDS.DAT
```

NOTE

> If you enter the *lfilename* specification, it must be complete. The filename and file extension from *rfilename* are *not* used as defaults for *lfilename*.

5.2.10 HELP Command

Use the HELP command to display a list of UFTP commands or help for a specific command.

Format

HELP [*command*]

Parameter

command

Identifies the command for which you want help.

5.2.11 LS Command

Use the LS command to list the files in a remote directory.

Format

LS [*remote-directory* [*lfilename*]]

Parameters

remote-directory

A remote directory specification.

If you omit the *remote-directory*, the current remote working directory is listed.

lfilename

A local file specification. This file receives the directory listing from the remote host.

If you omit *lfilename*, a listing of all files in the remote directory displays on your terminal.

Examples

The following command displays on your terminal a list of files in the current working directory of the remote host:

```
LS
```

The following command transfers a list of filenames from the current working directory of the remote host to the local file LIST.TXT:

```
LS "" LIST.TXT
```

You must use two double quotes to specify a null parameter.

5.2.12 MDELETE Command

Use the MDELETE command to delete multiple files from a remote host. If prompting is enabled, you are prompted to confirm the deletion of each file. Refer to the PROMPT command for more information.

Format

MDELETE *rfilenames*

Parameter

rfilenames

Remote file specifications. Usually, an asterisk (*) is used as a wildcard.

Examples

This example deletes all .TXT files from the current remote directory:

```
MDELETE *.TXT
```

This example deletes all files from the current remote directory:

```
MDELETE *.*
```

This example makes SMITH.MEMOS the current directory on the remote host, then deletes all .TXT files from that directory:

```
CD [SMITH.MEMOS]
MDELETE *.TXT
```

5.2.13 MGET Command

Use the MGET command to transfer multiple files from the remote host to your current local directory. If prompting is enabled, you are prompted to confirm the transfer of each file. Refer to the PROMPT command for more information.

Format

MGET *rfilenames*

Parameter

rfilenames

Remote file specifications. Usually, an asterisk (*) is used as a wildcard.

Examples

This example transfers all .TXT files from the current directory on the remote host to the current directory on the local host:

```
MGET *.TXT
```

This example transfers all files from the current directory on the remote host to the current directory on the local host:

```
MGET *.*
```

This example transfers all .TXT files from the [SMITH.MEMOS] directory on the remote host to the current directory on the local host:

```
CD [SMITH.MEMOS]
MGET *.TXT
```

5.2.14 MKDIR Command

Use the MKDIR command to create a directory on the remote host.

Format

MKDIR *directory-name*

Parameter

directory-name

This is the name of the directory you are creating.

Example

This example creates the directory MEMOS:

```
MKDIR MEMOS
```

5.2.15 MPUT Command

Use the MPUT command to transfer multiple files from your current directory on the local host to the current directory on the remote host. If prompting is enabled, you are prompted to confirm the transfer of each file. Refer to the PROMPT command for more information.

Format

MPUT *lfilenames*

Parameter

lfilenames

Local file specifications that identify the files being transferred to the remote host.

Examples

This example transfers all .TXT files from the current directory on the local host to the current directory on a remote host:

```
MPUT *.TXT
```

This example transfers all files from the current directory on the local host to the current directory on the remote host:

```
MPUT *.*
```

This example transfers all .TXT files from the [200,200] directory on the local host to the current directory on the remote host:

```
MPUT [200,200]*.TXT
```

5.2.16 OPEN Command

Use the OPEN command to open a connection to a remote host.

You must open a connection to a remote host before any data can be transferred.

If a connection is already open, the current connection is closed before a new one is opened.

Once the connection is opened, you are prompted for your username and password on the remote host. The password is not displayed.

If you fail to log in (for example, your username and password are not accepted by the remote host), the connection remains open. You can then use the USER command to send new login information to the remote host.

If the remote host does not support multi-user protection features, any username and password you type is acceptable.

The connection remains open until you exit UFTP, close the connection with the CLOSE command, or try to open a new connection.

Format

OPEN *host-name*

You can use CONNECT as a synonym for OPEN.

Parameter

host-name

This is the name or internet address of a remote host on the network. Upper and lowercase characters are equivalent. Refer to the *TCPware for RSX Software Reference Manual, Overview of TCP/IP Networking* chapter for information on host names.

Example

The following commands open a connection to the remote host ALPHA, get the remote file NOTES.TXT, and close the connection.

```
OPEN ALPHA
Username ? SMITH
Password ? {not echoed}
GET NOTES.TXT
CLOSE
```

5.2.17 PUT Command

Use the PUT command to transfer a file from your local host to the remote host.

Format

PUT *lfilename* [*rfilename*]

You can use SEND as a synonym for PUT.

See the TYPE command for information on file transfer representations.

Parameters

lfilename

A local file specification. This identifies the local file being sent to the remote host.

rfilename

A remote file specification. This is the name by which the file is stored on the remote host.

If you omit *rfilename*, the local file specification is also used as the remote file specification.

You must specify *rfilename* if the local file specification does not conform to the remote host's file naming conventions.

Example

The following command transfers the file RECRDS.DAT from the local host to the remote file RECORDS.DAT:

```
PUT RECRDS.DAT RECORDS.DAT
```

NOTE

If you enter the *rfilename* specification, it must be complete. The filename and file extension from *lfilename* are *not* used as defaults for *rfilename*.

5.2.18 PROMPT Command

Use the PROMPT command to toggle prompting for the MGET, MPUT, and MDELETE commands. This command helps prevent the wrong files from being transferred or deleted. The initial setting is on.

When prompted, you can respond as shown in the following table.

Response	Meaning
N	Do not perform the operation.
Y	Perform the operation.
G	Perform the operation on this file and all files that follow.
Q	Quit. Return to the command prompt.

If the input file specification that displays is not appropriate to use as the output file specification, the MGET and MPUT commands allow you to enter an output file specification. The output file specification must contain more than one character.

Prompting is disabled when UFTP is executing from a command file.

Format

PROMPT

5.2.19 PWD Command

Use the PWD command to display the name of the current working directory on the remote host.

Format

PWD

5.2.20 QUIT Command

Use the QUIT command to exit the UFTP utility and return to the DCL/MCR prompt.

Format

QUIT

If a connection is open, it will be closed before exiting.

You can use BYE and EXIT as synonyms for QUIT.

NOTE

The UFTP utility exits with status (using the EXST$S directive). It returns the EX$SUC status code only if there are no errors whatsoever. If any error occurred during operation, the EX$ERR status code is returned. This status information is useful primarily to indirect command files.

5.2.21 QUOTE Command

Use the QUOTE command to send any FTP protocol command to the remote server.

Format

QUOTE *string*

NOTE

The QUOTE command must never be used to initiate a file transfer.

Parameter

string

The File Transfer Protocol command string to send to the remote host. This should be enclosed within quotes.

Some useful commands are:

- **ACCT** *name* to specify account information, if required
- **ALLO** *bytes* to specify the file size, if required
- **SITE** to issue site-specific commands
- **SYST** to display the system type

Remote servers may not support all of the above commands.

Refer to the documents listed in the *References* section of the *General Information* chapter for information on the File Transfer Protocol commands.

Example

The following command sends the SYST command to the remote server. This command, if implemented by the remote server, returns the type of operating system on which the remote server is running.

```
QUOTE "SYST"
```

5.2.22 REMOTEHELP Command

Use the REMOTEHELP command to display help on the remote server.

Some servers may not support this command.

Format

REMOTEHELP [*topic*]

Parameter

topic

Identifies the topic on which you want help.

Some servers may not allow you to specify a topic.

5.2.23 RENAME Command

Use the RENAME command to rename a file on the remote host.

Format

RENAME *from-filename* *to-filename*

Parameters

from-filename

Specifies the name of the remote file you are renaming.

to-filename

Specifies the new name of the remote file.

Example

The following command renames the file OLD.TXT to NEW.TXT:

```
RENAME OLD.TXT NEW.TXT
```

5.2.24 RMDIR Command

Use the RMDIR command to delete a directory on a remote host.

Format

RMDIR *directory-name*

Parameter

directory-name

Specifies the name of the directory you are deleting.

Example

The following command deletes the MEMOS directory from the remote host:

```
RMDIR MEMOS
```

5.2.25 STATUS Command

Use the STATUS command to display the following information about your current session:

- UFTP version number
- name of the remote host if a connection is established
- file transfer representation type: ASCII, binary, formatted binary, FORTRAN, or default

Format

STATUS

5.2.26 TYPE Command

Use the TYPE command to change the file transfer representation type, or to define your own type.

The representation type describes how the data is represented when transferred across the network. UFTP supports the FTP ASCII Non-print, FTP ASCII Carriage Control, and FTP IMAGE types.

The FTP ASCII Non-print type is defined as ASCII data records terminated by the carriage return/line feed (CR/LF) character sequence. Use this type for all text and source files. Typical file extensions are .TXT, .FTN, and .MAC. Use the ASCII or TYPE ASCII command to select this mode. This is also the default mode when a connection is first opened.

The FTP ASCII Carriage Control type is almost identical to the FTP ASCII Non-print type, except the first character of each record is considered to be the ASA (FORTRAN) vertical format control character. Use the TYPE FORTRAN command to select this mode.

The FTP IMAGE type is defined as contiguous 8-bit bytes. Use binary type for .SAV, .TSK, .OLB, .MLB, .SYS, .SML, and .ULB files. Use the BINARY or TYPE BINARY command to select this mode.

The FTP IMAGE type is also used to transfer formatted binary data. This data includes the file data plus record structure information. This record structure information allows .OBJ, .STB, .BIN, and .LDA files to be transferred among systems. Use the TYPE FORMATTED_BINARY command to select this mode.

The remote server determines what format or record structure will be used to store a file. The remote server will do any necessary conversions between the FTP representation type and the local storage format.

Some remote servers may store the file differently depending on the file's extension. For example, Process Software Corporation's FTP-VMS software product stores or retrieves files transferred using the FTP IMAGE type as standard formatted binary (variable-length records) files if the file extension is .OBJ, .STB, .LDA, or .BIN. All other files transferred using FTP IMAGE type are stored as fixed-length 512-byte record files.

Format

TYPE *rep* [*mode,rtype*[,*rsize*[,*ratt*]]]

Parameters

rep

This must be one of the following keywords: ASCII, BINARY, FORMATTED_BINARY, FORTRAN, USER, or DEFAULT.

The keyword ASCII specifies that the ASCII representation type will be used for all future file transfers. This is the default type when a connection is first opened. The TYPE ASCII command has the same effect as the ASCII command.

The keyword BINARY specifies that the binary representation type will be used for all future file transfers. The TYPE BINARY command is the same as the BINARY command.

The keyword FORMATTED_BINARY specifies that the formatted binary representation type will be used for all future file transfers.

The keyword FORTRAN specifies that the ASCII Carriage Control representation type will be used for all future file transfers.

The keyword USER indicates that you are defining your own file type. The *mode* and *rtype* parameters are required with USER. The *rsize* and *ratt* parameters are optional.

The keyword DEFAULT specifies that UFTP determines the type for all future file transfers based on the local file's extension. BINARY is used if the file extension is .SAV, .TSK, .OLB, .MLB, .SYS, .SML, or .ULB. FORMATTED_BINARY is used if the file extension is .OBJ, .STB, .BIN, or .LDA. ASCII is used for all others.

mode

Defines the file transfer mode as ASCII or IMAGE. This is required when the *rep* is USER.

rtype

Defines the records as FIX (fixed) or VAR (variable) length. This is required when the *rep* is USER.

rsize

Defines the record size in bytes. You can use *rsize* only when the *rep* is USER. It is always optional.

ratt

Gives the record one of the attributes listed in the following table.

Attribute	**Meaning**
FTN	FORTRAN carriage control
CR	Carriage return carriage control
BLK	Non-spanned records

You can use *ratt* only when USER is the *rep*. It is always optional.

Examples

The following commands transfers a binary file, then change the representation type back to ASCII after the transfer is completed:

```
TYPE BINARY
GET MYPROG.SAV
TYPE ASCII
```

The following command defines a file type as ASCII, fixed-length, 128 bytes, carriage return carriage control:

```
TYPE USER ASCII,FIX,128,CR
```

5.2.27 USER Command

Use the USER command to send new login information to the remote host.

Format

USER *username password*

The user command allows you to log in to a different account on the same remote host without closing the connection.

If you omit the required parameters, you are prompted for them.

Let UFTP prompt you for the password so that it will not be echoed to your terminal.

Parameters

username

Your username on the remote host.

password

Your password on the remote host.

NOTE

> To specify a null username or password, you must use two double quotes (see the DIR command).

Example

The following example shows you how to transfer files from different accounts (USER1 and USER2) on the same remote host (SYSTEM1) using the USER command.

```
OPEN SYSTEM1
User name? USER1
Password? {not echoed}
{transfer files...}
USER
User name? USER2
Password? {not echoed}
{transfer files...}
CLOSE
```

5.3 UFTP Messages

UFTP can issue informational and error messages. These messages are self-explanatory.

Errors reported by the remote server are also given. These error messages have a three-digit error number followed by a text message.

UFTP displays all reply messages received from the Server-FTP.

NOTE

If the IE.ONP (Hardware/Software option not present) error message is issued by UFTP, the software password is invalid for the current configuration. Ask the system manager whether the software password was entered correctly.

5.4 Aborting Commands and File Transfers

You can abort the DIR, GET, LS, MGET, MPUT, OPEN, and PUT commands by typing ^C.

If the OPEN command is aborted, no connection will be open.

If the DIR, GET, LS, MGET, MPUT, or PUT command is aborted, both the data and control connections are aborted. If you want to transfer more files, you will have to open a connection and log in to the remote host again.

NOTE

The STATUS command may incorrectly indicate that the connection to the remote host is still open.

5.5 Using Indirect Command Files

UFTP can use command files; however, they must be invoked from within the utility. You invoke a UFTP command file by typing:

`@`*command-file*

where *command-file* is a text file that contains UFTP commands (one per line). Command files can be nested to a depth of 3.

The following example shows you how to use indirect command files and UFTP command files together to automate the process of transferring files.

Two command files are used in this example.

The file XFER.CMD is an indirect command file. XFER.CMD contains:

```
UFTP @SUB
TYPE NOTES.TXT
```

The file SUB.CMD is a UFTP command file. SUB.CMD contains:

```
OPEN SYSTEM1
SMITH SECRET
GET NOTES.TXT
QUIT
```

Invoke the XFER.CMD file by typing (at the DCL/MCR prompt):

```
@XFER
```

This command receives the file NOTES.TXT from the remote host SYSTEM1, and then displays the contents of the file on your terminal.

NOTE

If you want to supply the password from the command file, you must enter it on the same line as the username. If the password is not supplied with the username, UFTP prompts you for the password from the terminal.

5.6 Library Routine Usage

This section describes how the UFTP utility uses FTP-RSX library routines. The material in this section is provided for users of the FTP-RSX library routines and is not intended for most readers.

The following list describes the FTP-RSX library routines used by each UFTP command:

- ASCII, BINARY, and TYPE commands use the FTPSFT routine.
- CD command uses the FTPCWD routine.
- CLOSE command uses the FTPCCL routine.
- DELETE and MDELETE commands use the FTPDEL routine.
- DIR command uses the FTPDIR routine.
- GET and MGET commands use the FTPGET routine.
- LS, MDELETE, and MGET commands use the FTPDNL routine.
- OPEN command uses the FTPOOC and FTPLIN routines.
- PUT and MPUT commands use the FTPPUT routine.
- QUIT command uses the FTPCCL routine if a connection is open.
- QUOTE, REMOTEHELP, MKDIR, RMDIR, RENAME, and PWD commands use the FTPQUO routine.
- USER command uses the FTPLIN routine.

The FTPINI routine is called during UFTP initialization. The FTPABO routine is called whenever a ^C is typed.

6
FTP Library Reference

This chapter describes the FTP-RSX library routines.

FTP-RSX library routines are in the FTPLIB.OLB object library.

The FTP-RSX library includes the following routines:

- **FTPABO** - Abort operation
- **FTPCCL** - Close connection
- **FTPCWD** - Change working directory
- **FTPDEL** - Delete file
- **FTPDIR** - Returns a directory listing
- **FTPDNL** - Returns a file listing
- **FTPGET** - Transfer file from remote to local
- **FTPINI** - Initialization
- **FTPLIN** - Send login information
- **FTPOOC** - Open an FTP connection
- **FTPPUT** - Transfer file from local to remote
- **FTPQUO** - Send FTP command
- **FTPSFT** - Set file transfer information

All routines use the standard PDP-11 R5 calling convention. All routines should be called as integer functions because a status code is returned.

Status codes indicate the outcome of the operation. The symbolic names for the status codes are defined in the FTP-RSX library routines object library. Positive values indicate success. Negative values indicate errors.

See the *Status Codes* section later in this chapter.

A sample program that uses the FTP-RSX library routines is in the *Sample Program* appendix.

NOTE

> When writing application programs, use only the library routines that are documented in this guide. The FTP-RSX library uses additional global symbols for internal routines. These symbols are named FT*xxxx,* to reduce the chances of conflict with other symbol names.

6.1 Library Routines

This section describes each FTP-RSX library routine in detail.

6.1.1 FTPABO Routine

Call the FTPABO routine to abort the current operation (if any). This routine aborts only the FTPDIR, FTPDNL, FTPGET, FTPOOC, and FTPPUT subroutines.

Format

ret-status = FTPABO(*cid*)

Argument

cid

Is the connection id. This integer value is currently ignored.

Notes

If the current operation is aborted, all connections to the remote host are closed.

This routine is usually used in conjunction with an AST routine to abort file transfers in response to a ^C or some other event.

FTPABO can be called at any time after FTPINI.

6.1.2 FTPCCL Routine

Call the FTPCCL routine to close the FTP connection.

Format

ret-status = FTPCCL(*cid*)

Argument

cid

Is the connection id. This integer value is currently ignored.

Notes

FTPCCL waits until the connection is closed.

FTPCCL uses the FTP command QUIT.

6.1.3 FTPCWD Routine

Call the FTPCWD routine to change the remote working directory.

Format

ret-status = FTPCWD(*cid*,*rdir*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

rdir

Is the ASCIZ string containing the remote directory specification. The maximum allowable length of this string is 132 characters.

Notes

FTPCWD may be called only while a connection is open.

If the directory string is null, the working directory is changed to the parent directory.

FTPCWD uses the FTP command CWD (or CDUP if *rdir* is a null string).

6.1.4 FTPDEL Routine

Call the FTPDEL routine to delete a remote file.

Format

ret-status = FTPDEL(*cid*,*rfilename*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

rfilename

Is the ASCIZ string containing the remote file specification of the file to delete. The maximum allowable length of this string is 132 characters.

Notes

FTPDEL may be called only while a connection is open.

FTPDEL uses the FTP command DELE.

6.1.5 FTPDIR Routine

Call the FTPDIR routine to receive the text of a remote directory listing. Depending on the server, this list might include the name, size, creation date, and protection information associated with the file.

Format

ret-status = FTPDIR(*cid*,*lfilename*,*rdir*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

lfilename

Is the ASCIZ string containing the local file specification of a file to receive the remote directory listing. The *lfilename* must be a valid RSX file specification. If this is a null string, the directory listing is displayed on the terminal (TI:).

rdir

Is the ASCIZ string containing the remote directory specification of the directory to list. The maximum allowable length of this string is 132 characters.

Notes

FTPDIR may be called only while a connection is open.

FTPDIR uses the FTP command LIST.

6.1.6 FTPDNL Routine

Call the FTPDNL routine to receive the text of a remote directory listing. This listing contains only the names of the files in the directory.

Format

ret-status = FTPDNL(*cid*,*lfilename*,*rdir*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

lfilename

Is the ASCIZ string containing the local file specification of a file to receive the remote directory listing. The *lfilename* must be a valid RSX file specification. If this is a null string, the directory listing is displayed on the terminal (TI:).

rdir

Is the ASCIZ string containing the remote directory specification of the directory to list. The maximum allowable length of this string is 132 characters.

Notes

FTPDNL can be called only while a connection is open.

FTPDNL uses the FTP command NLIST.

6.1.7 FTPGET Routine

Call the FTPGET routine to receive a file from the remote host.

Format

ret-status = FTPGET(*cid*,*lfilename*,*rfilename*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

lfilename

Is the ASCIZ string containing the local file specification of the file that will receive the remote file.

Argument *lfilename* should be a valid RSX file specification.

rfilename

Is the ASCIZ string containing the remote file specification of the remote file to be received. The maximum allowable length of this string is 132 characters.

Notes

FTPGET may be called only while a connection is open.

FTPGET uses the FTP command RETR.

6.1.8 FTPINI Routine

Call the FTPINI routine to do the one-time initialization of the FTP-RSX library routines. This process initializes various data structures.

FTPINI must be called before any other FTP-RSX library routines. FTPINI returns an error if it is called more than once.

Format

ret-status = FTPINI(*baslun*, *basefn*)

Arguments

baslun

Is a word containing the first available logical unit number (LUN). Up to four logical units are used by the FTP-RSX library routines (*baslun*, *baslun*+1, *baslun*+2, and *baslun*+3).

basefn

Is a word containing the first available event flag number. Up to four event flags may be used (*basefn*, *basefn*+1, *basefn*+2, *basefn*+3). All four event flags MUST be in the same event flag group.

6.1.9 FTPLIN Routine

Call the FTPLIN routine to send login information to the remote host you are currently connected to.

Format

ret-status = FTPLIN(*cid*,*username*,*password*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

username

Is the ASCIZ string containing your username on the remote host. The maximum allowable length of this string is 132 characters.

password

Is the ASCIZ string containing your password on the remote host. The maximum allowable length of this string is 132 characters.

The password is sent only if requested by the remote host.

Notes

FTPLIN may be called only while a connection is open.

Some hosts may not support a change in username or password after login.

FTPLIN uses the FTP commands USER and PASS.

6.1.10 FTPOOC Routine

Call the FTPOOC routine to open a connection to a remote host. A connection must be opened before any data can be transferred.

Format

ret-status = FTPOOC(*cid*,*host)*

Arguments

cid

Is the connection id. This is a word value returned to identify the connection.

The *cid* is passed to all the FTP-RSX library routines that send or receive data across a connection.

Connection id's are not currently used. No value is returned by FTPOOC and the *cid* argument is ignored by all other routines. This argument is reserved for future use.

host

Is the ASCIZ string containing the host name or internet address string of the host you want to connect to. The maximum allowable length of this string is 32 characters.

Notes

FTPOOC must be called after FTPINI and before FTPLIN, FTPSFT, FTPCWD, FTPDEL, FTPDIR, FTPGET, FTPPUT, FTPQUO, FTPCCL, or FTPDNL. FTPOOC does not return until the connection has opened or an error has occurred.

6.1.11 FTPPUT Routine

Call the FTPPUT routine to send a file to the remote host.

Format

ret-status = FTPPUT(*cid*,*lfilename*,*rfilename*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

lfilename

Is the ASCIZ string containing the local file specification of the file to be sent to the remote host.

Argument *lfilename* should be a valid RSX file specification.

rfilename

Is the ASCIZ string containing the remote file specification of the remote file that will receive the local file. The maximum allowable length of this string is 132 characters.

Notes

FTPPUT may be called only while a connection is open.

FTPPUT uses the FTP command STOR.

6.1.12 FTPQUO Routine

Call the FTPQUO routine to send a File Transfer Protocol (FTP) command to the remote server.

Format

ret-status = FTPQUO(*cid*,*command*)

Arguments

cid

Is the connection id. This integer value is currently ignored.

command

Is the ASCIZ string containing the FTP command to be sent. Any FTP command not requiring a data connection can be used. The maximum allowable length of this string is 132 characters.

Notes

FTPQUO may be called only while a connection is open.

FTPQUO must not be used to send commands that use a data connection (the command must not initiate a file transfer).

6.1.13 FTPSFT Routine

Call the FTPSFT routine to set file transfer information.

Format

ret-status = FTPSFT(*cid*,*typeflag*[,*struflag*[,*filename*]])

Arguments

cid

Is the connection id. This integer value is currently ignored.

typeflag

Is a word value used to specify the file transfer representation type.

A value of 1 selects FTP ASCII, 2 selects formatted binary (FTP IMAGE), 4 selects image (FTP IMAGE), 8 selects FORTRAN carriage control (FTP ASCII Carriage Control), and a value of 0 selects the type based on the argument *filename*'s file extension (if specified).

struflag

Is a word value. Argument *struflag* is not currently supported and is ignored.

struflag is an optional parameter.

filename

Is the ASCIZ string containing the file specification used to determine the representation type if *typeflag* is 0.

Argument *filename* should be a valid RSX file specification.

Argument *filename* is an optional parameter and is used only if *typeflag* is 0.

Notes

FTPSFT may be called only while a connection is open.

FTPSFT uses the FTP command TYPE.

6.2 Status Codes

The FTP-RSX library routines can return the following status codes:

Symbol	Value	Description
IS.SUC	1	Success.
SC.RNI	-17	Root is not initialized (call FTPINI first).
SC.CNC	-18	Connection is not closed (close previous connection first).
SC.CNO	-19	Connection is not open (open connection first).
SC.CID	-21	Connection is deleted (from abort).
SC.BNP	-50	Bad parameter or number of parameters.
SC.IID	-58	Initialization is already done.

Error codes from -100 to -199 and -300 to -599 are FTP server reply codes from the remote host. These codes indicate that the operation did not complete successfully.

Error codes from -1001 to -1127 are RSX I/O errors -1 to -127, respectively.

Error codes from -2001 to -2127 are RSX directive status (DSW) errors -1 to -127, respectively.

Values from 200 to 299 are FTP server reply codes that indicate success.

6.3 Linking with the Library

The FTP-RSX library routines are provided in the FTPLIB.OLB file. FTPLIB.OLB is an object library file, which consists of the modules required to use the FTP-RSX library routines.

To link your application program with the FTP-RSX library routines, simply include the LB:[1,1]FTPLIB.OLB/LB library specification in the list of object files used by the program.

6.4 Debugging Support

The FTP-RSX library routines (and User-FTP utility) include debugging support that can be enabled to help isolate the source of communication problems.

To enable debugging support, modify the FTDBGF location as described below.

The bit mask values (in octal) for the FTDBGF location are:

Mask	Meaning (if set)
000020	Display received replies
000040	Display transmitted commands

Set the bits in the FTDBGF location corresponding to the information desired. The address of the FTDBGF location can be found from the link map. UFTP always sets the FTDBGF location with mask 000020 to display received replies.

You can set the bits in one of three ways: have your program write to the location, use the ZAP utility, or use the TKB GBLPAT option.

7 Server-FTP (FTPD)

This chapter describes the Server-FTP (FTPD) task provided with FTP-RSX.

The Server-FTP (FTPD) task services a connection initiated by a Client-FTP utility (such as FTP-RSX's UFTP). Each server can service one connection. To service many connections simultaneously, you must start multiple Server-FTP (FTPD) tasks. When a server finishes servicing a Client-FTP, it listens for another connection.

The Server-FTP (FTPD) task runs unattended as a privileged task. It does not require operator input.

7.1 FTP Commands

The FTP-RSX Server-FTP (FTPD) supports the following File Transfer Protocol commands:

ALLO	LIST	QUIT	REIN	SYST
CDUP	MODE	PASS	RETR	TYPE
CWD	NLST	PORT	STOR	USER
DELE	NOOP	PWD	STRU	

The *File Transfer Protocol*, RFC-959 from the Information Sciences Institute at the University of Southern California, provides more information on these commands.

Implementation-specific information on these commands is given in the following paragraphs. Also refer to the *Implementation Notes* appendix.

For the ALLO command, only the first argument is used. This argument is the number of bytes to be allocated for the next file (it is rounded up to the number of blocks). If the value is negative, a contiguous allocation is made.

For the LIST and NLST commands, wildcard directory specifications are not supported. The file name, file type, and version number may be wild. However, only the wildcard syntax supported by the FCS .FIND routine is supported (see the DEC *I/O Operations Reference Manual).*

For the MODE command, only the stream (S) parameter is supported. For the STRU command, only the file (F) parameter is supported. For the TYPE command, the I, A, A N, and A C parameters are supported.

For the PASS and USER commands, the password and username are restricted to the lengths supported by RSX. On systems without the multi-user protection support, these commands are not required, and any username is accepted. On systems with the multi-user protection support, the username and password are validated by reading the accounting file; the defaults and protection information for the account are used. If logins are disabled, a remote user cannot log in.

NOTE

> The USER command supports only usernames. It does not support UICs.

FTP-RSX supports the following qualifiers for the RETR and STOR commands:

- /ASCII - Indicates the file is read or written as an ASCII file.
- /BINARY - Indicates the file is read or written as a formatted binary file. This qualifier is useful when transferring variable length binary files that do not have a file extension of .OBJ, .STB, .BIN, or .LDA.
- /FORTRAN - Indicates the file is read or written as a FORTRAN carriage control file.
- /IMAGE[:*n*] - Indicates the file is read or written as an image file. If a record length is specified, it only applies to output files.

7.2 Site-Specific Command

You can enter a site-specific command from any FTP client that is connected to the FTP-RSX server.

The server supports one site-specific command: SITE TYPE USER *mode,rtype,*[*rsize*[*,ratt*]

This command allows you to define file types on the FTP-RSX server.

Parameters

mode

Defines the file transfer mode as ASCII or IMAGE. Required.

rtype

Defines the records as FIX (fixed) or VAR (variable) length. Required.

rsize

Defines the record size in bytes. Optional.

ratt

Assigns the record one of the attributes listed in the following table.

Attribute	Meaning
FTN	FORTRAN carriage control
CR	Carriage return carriage control
BLK	Non-spanned records

The *ratt* parameter is optional.

Example

The following server command defines a file type as ASCII, fixed-length, 128 bytes, carriage return carriage control:

```
SITE TYPE USER ASCII,FIX,128,CR
```

7.3 Event Logging Subroutine

This section describes writing an optional event logging subroutine. The event logging subroutine you write is linked with the FTPD task. It is called whenever one of the following events occurs:

- Control connection opened (EV.CCO),
- Control connection lost (EV.CCL),
- Control connection closed (EV.CCC),
- User logged in on connection (EV.LIN),
- File deleted by user (EV.DEL),
- File sent to remote system (EV.GET), and
- File received from remote system (EV.PUT).

The EV.xxx symbols are the global symbols your event logging subroutine defines for the events.

When one of the events occurs, the event logging subroutine is called. It is passed the event number and the address of the connection control block (CCB). The CCB contains all of the information about a connection (including the FDB, username, user context information, and current context settings). The template logging subroutine module, LOGSUB.MAC, provided with the distribution kit, describes the method in which the event number and CCB are passed. The fields within the CCB are also described in this module. Please see this file for more information.

The event logging subroutine is free to do whatever it likes. However, it must preserve all registers and should complete quickly. It is recommended that any processing of the information be done in another task, and that the event logging subroutine send that task a message (using the RSX intertask communication directives).

There are two source files provided with FTP-RSX related to event logging. The template event logging subroutine, in the file LOGSUB.MAC, builds a data buffer (it extracts pertinent information for the event from the CCB) and then uses the VSDA$S directive to send this data buffer to the FTPD.L task (any errors returned by the VSDA$S directive are ignored). The template logging task, in the file FTPLOG.MAC, receives these data buffers and displays the information on the terminal (the device assigned to LUN 5). The task should be installed and run as FTPD.L. Please see the source files for more details on their operation.

NOTE

> The LOGSUB.MAC and FTPLOG.MAC files, as provided, cannot be used under RSX-11M because these files use the send and receive variable data directives. RSX-11M does not support these directives.

The installation procedure prompts for the name of the event logging subroutine module. If you do not want to use this feature of the Server-FTP, simply press return or enter NULLOG. The NULLOG module contains a null event logging subroutine and should be used whenever event logging is not required.

A
Sample Program

The following sample program shows you how the FTP-RSX library routines can be used in your own programs. The sample program is in FORTRAN. This program is supplied with the distribution kit in the SAMPLE.FTN file.

The program prompts you for the host name, username, password, transfer mode (ASCII or binary), operation (send or receive), and local and remote filenames. The file is then transferred.

```
        PROGRAM SAMPLE
C
C This program demonstrates how the FTPLIB routines can be called.
C It can be used to transmit or receive a file from a remote host.
C The program prompts for all of the required information.
C
C
        EXTERNAL  FTPINI,FTPOOC,FTPLIN,FTPSFT,FTPGET,FTPPUT,FTPCCL
        INTEGER*2 FTPINI,FTPOOC,FTPLIN,FTPSFT,FTPGET,FTPPUT,FTPCCL

        BYTE LFN(41), RFN(65), UN(33), PW(33), SYS(33)
        INTEGER*2 DIRECT, STATUS, MODE, CID

1       TYPE *,'FTP-RSX Library -- Sample Program'

        TYPE *,'Initializing the FTP library routines'
        STATUS = FTPINI(1,1)
        TYPE *,'FTPINI status is ',STATUS
        IF (STATUS .LT. 0) GOTO 990

        TYPE *,'Host name ?'
        CALL GETSTR(SYS,32)
        STATUS = FTPOOC(CID,SYS)
        TYPE *,'FTPOOC status =',STATUS
        IF (STATUS .LT. 0) GOTO 990

        TYPE *,'User name ?'
        CALL GETSTR(UN,32)
```

```
        TYPE *,'Password ?'
        CALL GETSTR(PW,32)
        STATUS = FTPLIN(CID,UN,PW)
        TYPE *,'FTPLIN status =',STATUS
        IF (STATUS .LT. 0) GOTO 990

        TYPE *,'Transfer mode (1,2,4,8) ?'
        ACCEPT *,MODE
        STATUS = FTPSFT(CID,MODE)
        TYPE *,'FTPSFT status =',STATUS
        IF (STATUS .LT. 0) GOTO 990

        TYPE *,'Operation (0 = Send file, 1 = Receive file) ?'
        ACCEPT *,DIRECT
        IF ((DIRECT .NE. 0) .AND. (DIRECT .NE. 1)) THEN
          TYPE *,'No file sent or received.'
          GOTO 990
        END IF
        TYPE *,'Local file name ?'
        CALL GETSTR(LFN,40)
        TYPE *,'Remote file name ?'
        CALL GETSTR(RFN,64)
        IF (DIRECT .EQ. 0) THEN
          STATUS = FTPPUT(CID,LFN,RFN)
          TYPE *,'FTPPUT status =',STATUS
        ELSE
          STATUS = FTPGET(CID,LFN,RFN)
          TYPE *,'FTPGET status =',STATUS
        END IF

990     STATUS = FTPCCL(CID)
        TYPE *,'FTPCCL status =',STATUS

        END

        SUBROUTINE GETSTR(INVEC,NUM)
C This subroutine obtains from the user an ASCIZ input string.
        INTEGER*2 I, NUM
        BYTE INVEC(NUM+1)
        ACCEPT 10, INVEC
10      FORMAT (80A1)
        I = NUM
20      IF ((INVEC(I) .NE. ' ') .OR. (I .EQ. 0)) GOTO 30
        I = I-1
        GOTO 20
30      INVEC(I+1) = 0
        END
```

B
Implementation Notes

This appendix describes the FTP-RSX implementation of the File Transfer Protocol (FTP). The material in this appendix requires a thorough understanding of the protocols used and is not intended for most readers.

For a description of the implementation of the Transmission Control (TCP), Internet (IP), and Address Resolution (ARP) protocols, refer to the *TCPware for RSX Software Reference Manual*.

This appendix does not describe the File Transfer Protocols. Please refer to RFC-959 (see the *References* section of the *General Information* chapter) for the File Transfer Protocol specification.

B.1 File Transfer Protocol (FTP)

The following File Transfer Protocol service commands are implemented:

- **ALLOCATE** (ALLO *arguments*) - specifies file size information. It is not issued. The file size (byte count) is used if this command is received. A negative byte count can be specified to allocate a contiguous file.
- **CHANGE WORKING DIRECTORY** (CWD *directory*) - changes the working directory.
- **CHANGE TO PARENT DIRECTORY** (CDUP) - changes the working directory to the parent (login) directory.
- **DATA PORT** (PORT *arguments*) - specifies the data port number to be used for data transfers.
- **DELETE** (DELE *filespec*) - deletes the file(s) specified.
- **FILE STRUCTURE** (STRU *arguments*) - specifies the file structure. The only valid argument value is F (FILE = no record structure).
- **HELP** (HELP [*topic*]) - requests help information. The UFTP utility supports this command; the FTPD server does not support it.
- **LIST** (LIST *filespec*) - requests a directory listing to be sent.
- **LOGOUT** (QUIT) - closes the connection.

- **MAKE DIRECTORY** (MKD *directory*) - creates the specified directory. The UFTP utility supports this command; the FTPD server does not support it.
- **NAME LIST** (NLST *filespec*) - returns a list of file names.
- **NOOP** (NOOP) - this command is ignored; however, it is acknowledged with a success status.
- **PASSWORD** (PASS *password*) - specifies the user's password and logs the user into the system.
- **PRINT WORKING DIRECTORY** (PWD) - returns the current working directory.
- **REINITIALIZE** (REIN) - logs out the user and resets the file transfer parameters to the initial values.
- **REMOVE DIRECTORY** (RMD *filespec*) - deletes the specified directory. The UFTP utility supports this command; the FTPD server does not support it.
- **RENAME FROM** (RNFR *filespec*) - specifies the file to be renamed. The UFTP utility supports this command; the FTPD server does not support it.
- **RENAME TO** (RNTO *filespec*) - specifies the new name of the file designated in the RNFR command. The UFTP utility supports the RNTO command; the FTPD server does not support it.
- **RETRIEVE** (RETR *filespec*) - reads the file and transmits it to the requesting host.
- **REPRESENTATION TYPE** (TYPE *arguments*) - specifies the file type. The valid argument values are A (ASCII), I (image), N (non-print), and C (FORTRAN carriage control). I (image) is used for both formatted binary and image format transfers. Formatted binary data includes the necessary record headers and checksums.
- **SITE TYPE USER** - allows you to define file types on the FTP-RSX server from a remote client. Refer to the *Server-FTP (FTPD)* chapter for more information.
- **STORE** (STOR *filespec*) - writes the file from data received from the requesting host.
- **SYSTEM** (SYST) - requests the name of the operating system on which the server is running.
- **TRANSFER MODE** (MODE *arguments*) - specifies the transfer mode. The only valid argument value is S (STREAM).
- **USER NAME** (USER *name*) - specifies the user's username.

All other commands received by the server result in error 500 (command not implemented or syntax error).

The following response codes may be returned by the server: 150, 200, 215, 220, 221, 226, 230, 257, 331, 451, 500, 501, 504, 530, and 550. All response codes are accepted by UFTP and the FTP-RSX library.

B.2 Memory Usage

This table shows the approximate amount of memory each FTP-RSX software module requires. These sizes are rough estimates.

Module	Memory Requirements	
UFTP User Utility	26K bytes	(using FCSRES)
FTPD Server Task	22K bytes	(using FCSRES)
IN Device Driver	10K bytes	
INSACP Task	10K bytes	
XQ Device Driver	4K bytes	
XU Device Driver	6K bytes	
FTP-RSX Library	14K bytes	

C
Release Notes

This appendix summarizes the enhancements and changes for each version of FTP-RSX. If you have used earlier versions of FTP-RSX, read this appendix to determine whether any changes affect you.

Changes affecting the TCP-RSX software product are documented in the *TCPware for RSX Reference Manual, Release Notes* appendix.

C.1 Version 2.2-5

- No changes were made to the FTP-RSX components in the maintenance release.

C.2 Version 2.2-4

This section describes enhancements and changes for FTP-RSX Version 2.2-4.

- Server-FTP (FTPD) now handles the tail end of data transfers somewhat differently to prevent a deadlock condition:

  The data transfer completion status message is now sent out on the control connection before the data connection is closed. Previously, the data connection was closed first, which could cause problems for implementations expecting the completion status to be sent first.

- FTPD now sets the owner UIC to the default UIC for privileged accounts.

  If a privileged user changes his or her default UIC, the UIC for created files tracks that default UIC instead of maintaining the original login UIC. This matches what RSX does for privileged interactive users.

- The FTP-RSX file transfer information routine (FTPSFT) now sets the file transfer type correctly when called to determine it from a file's extension.

  User-FTP (UFTP) also now operates correctly in TYPE DEFAULT mode, since UFTP uses the FTPSFT routine for this mode.

- FTP-RSX now treats <CR><LF> *and* <LF> as line terminators for ASCII file transfers. Previously, FTP-RSX supported only <CR><LF>, which could cause problems with some nonconformant FTP implementations.

C.3 Version 2.2

This section describes enhancements and changes for FTP-RSX Version 2.2.

C.3.1 Commands

- The UFTP utility contains the following new commands: BELL, LS, MDELETE, MGET, MKDIR, MPUT, PROMPT, PWD, RMDIR, REMOTEHELP, and RENAME. The *User-FTP Utility* chapter describes these commands.
- The UFTP utility permits you to define additional RSX file types by means of the TYPE USER command. The *User-FTP Utility* chapter describes this command in detail.
- The FTPD server permits you to define additional RSX file types by means of the SITE TYPE USER command. Refer to the *Server-FTP (FTPD)* chapter for details on this command.
- The FTPD server supports the /ASCII, /BINARY, /FORTRAN, and /IMAGE[:*n*] qualifiers with the File Transfer Protocol RETR and STOR commands. This enhancement allows you to communicate more information about RSX file formats. Refer to the *Server-FTP (FTPD)* chapter for details.

C.3.2 Library Routines

- All known problems with the FTP-RSX library routines are corrected. In addition, the number of global symbols has been reduced, resulting in fewer chances for symbol names to conflict with existing applications. Most internal global symbols are named FT*xxxx*.
- FTP-RSX supports a new library routine, FTPDNL. Use FTPDNL to receive the text of a remote directory listing. Refer to the *FTP Library Reference* chapter for more information.

C.3.3 Additional Improvements

- FTP-RSX uses smaller receive windows for the FTP control connection, and tries to use larger windows for the FTP data connection.
- A potential problem affecting access of the RSX accounting file is corrected. If a user tries to log into the FTPD server while the accounting file is in use, FTPD waits a short time and tries to open the file again. It repeats the attempt several times, if necessary.
- Additional problems that the UFTP utility and FTPD server had in previous versions are corrected.

C.4 Version 2.1

This version includes the following several enhancements and changes:

- Minor corrections have been made to the UFTP utility and FTPD server.
- The *TCPware for RSX Software Reference Manual* now contains some of the material previously included in this manual.
- The CONNECT and DISCONNECT synonyms have been added for the OPEN and CLOSE commands respectively.
- The documentation has been updated to include the above enhancements.

Refer to the *Release Notes* appendix of the *TCPware for RSX Software Reference Manual* for information on the enhancements and corrections to the lower software layers.

C.5 Version 2.0

FTP-RSX V2.0 is a major rework of the FTP-RSX software product. It contains the following enhancements and changes:

- The protocols are now layered and can be used by several application programs simultaneously. New components (INDRV, INSACP, and NETCU) have been added to the product. See the *General Information* chapter.
- The FTP utility has been replaced with a new utility. This utility uses a UNIX FTP-like command syntax. It also uses the FTP-RSX library routines. See the *Transferring Files*, *User-FTP Utility*, and *FTP Library Reference* chapters.
- The FTP-RSX library routines have been added. Application programs can be written to use these subroutines to transfer files. See the *FTP Library Reference* chapter.
- The Server-FTP (FTPD) task now services only one remote connection. It also supports directory operations, encrypted passwords, and named directories. Multiple Server-FTP tasks can be run to service multiple remote connections.
- TCP services have been added. Application programs can now be written to send and receive data over TCP connections. See the *TCP Services* chapter in the *TCPware for RSX Software Reference Manual*.
- Software password protection has been added to FTP-RSX. You must contact Process Software Corporation during the installation procedure to get the software password for your system.
- The installation procedure has been revised considerably. See the *Installation* chapter.
- The documentation has been updated to include the above enhancements.

Index

A

Aborting commands and file transfers, 33, 36
Address Resolution Protocol (ARP), 51
ASCII command, 12, 19, 20, 34
  See also TYPE command

B

BELL command, 12, 19, 20
BINARY command, 12, 19, 20, 34
  See also TYPE command
BYE command. *See* QUIT command

C

CD command, 12, 19, 20, 34
CLOSE command, 10, 12, 19, 21, 26, 34, 57
Command files, 28, 33–34
Command parameters, 19
Communications protocols, 15
Components
  diagram, 16
  hardware, 15
  overview, 15–17
  software, 15
CONNECT command, 57
  See also OPEN command
Connection Control Block (CCB), 47
  CCB fields, 47
Contiguous file allocation, 45, 51
Control C, 33, 34, 36

D

DEBUG command, 21
DELETE command, 12, 19, 21–22, 34
Device drivers
  IN, 17
  network, 17
DIR command, 12, 19, 22, 32, 34
  aborting, 33
DISCONNECT command, 57
  See also CLOSE command
Documentation set, 5
Double quotes
  for command parameters, 19
  for null parameter, 22, 23
  for remote file specifications, 11

E

EXIT command. *See* QUIT command

F

File Transfer Protocol (FTP), 1, 15
File Transfer Protocol commands, 28, 45–46, 51–52
FTP-RSX, definition, 1
FTPABO routine, 34, 35, 36
FTPCCL routine, 34, 35, 36, 40
FTPCWD routine, 34, 35, 37, 40
FTPD. *See* Server-FTP task
FTPDEL routine, 34, 35, 37, 40
FTPDIR routine, 34, 35, 36, 37–38, 40
FTPDNL routine, 34, 35, 36, 38, 40
FTPGET routine, 34, 35, 36, 39, 40
FTPINI routine, 34, 35, 36, 39, 40, 42
FTPLIN routine, 34, 35, 40
FTPOOC routine, 34, 35, 36, 40
FTPPUT routine, 34, 35, 36, 40, 41
FTPQUO routine, 34, 35, 40, 41
FTPSFT routine, 34, 35, 40, 42

G

GET command, 10, 12, 19, 22–23, 34
  See also TYPE command
  aborting, 33

H

Hardware configuration, 15
HELP command, 12, 19, 23
Host Name Table, 9
Host names, 9, 26, 40, 49

I

Indirect command files, 33–34
INSACP, 17
Installation, 7
Internet addresses, 9, 40
Internet Protocol (IP), 1, 15, 51

L

Library routines, 4, 16, 17, 36–42, 57
  descriptions of, 35–45
  enabling debugging support, 43
  initialization of, 39
  linking with applications, 43
  list of, 35
  sample, 49–50
  status codes, 42–43
  used by UFTP commands, 34
LS command, 12, 19, 23, 34
  aborting, 33

M

MDELETE command, 13, 19, 24, 34
MDIR command, 13
Memory usage, 53
MGET command, 13, 19, 24, 34
  aborting, 33
MKDIR command, 19, 25, 34
MPUT command, 13, 19, 25, 34
  aborting, 33

N

NETCU. *See* Network Control Utility
Network Control Utility, 9, 17
  running, 9

O

OPEN command, 10, 13, 19, 26, 34, 57
  aborting, 33
Opening a connection, 10, 26

P

Password, remote host account, 9, 26
Password, software
  invalid, 33
  obtaining, 7
PROMPT command, 13, 19, 27
Protocols supported, 1
PUT command, 10, 13, 19, 26–27, 34
  See also TYPE command
  aborting, 33
PWD command, 13, 19, 27, 34

Q

QUIT command, 12, 13, 19, 28, 34
QUOTE command, 13, 19, 28, 34

R

R5 calling convention, 35
RECV command, 22
References, 18
Release notes, 55–56
REMOTEHELP command, 13, 19, 29, 34
RENAME command, 13, 19, 29, 34
RMDIR command, 13, 19, 29, 34

S

Sample program, 49–50
SEND command. *See* PUT command
Server command, site-specific, 46
Server-FTP task, 17, 45
  event logging subroutine, 47
SITE TYPE USER command, 46
Site-specific command, 46
Special symbols, 11
Status codes, 42–43
  after exiting UFTP, 28
  IS.SUC, 42
  SC.BNP, 42
  SC.CID, 42
  SC.CNC, 42
  SC.CNO, 42
  SC.IID, 42
  SC.RNI, 42

STATUS command, 13, 19, 30, 33
Status messages, 32–33
Symbols, global, 36

T

TCP-RSX, purpose, 2
TCPware for RSX, introduction to family, 1
TELNET-RSX, purpose, 2
Transferring files
  description, 9
  exiting UFTP, 12
  opening a connection, 10
  preparing for, 9
  running UFTP, 10
  sample, 13
  using GET and PUT commands, 10–12
Transmission Control Protocol (TCP), 1, 15, 51
TYPE command, 13, 19, 30–31, 34

U

UFTP. *See* User-FTP utility
USER command, 13, 19, 32–36
User Datagram Protocol (UDP), 1
User-FTP utility
  aborting, 33
  command files, 33
  commands, 19–32
  description, 16
  messages, 32–33
  sample, 13

W

Wildcards, 45