Connect:Express®
z/OS

FTP Guide

Version 4.2.3

Sterling Commerce
An IBM Company

Connect:Express z/OS FTP Guide
Version 4.2.3
First Edition

Printed in the United States of America.

CXZOSFG001

Contents

Preface

Chapter 1 Connect:Express and FTP Transfers

Introduction 1-1
The FTP Protocol 1-1

Connect:Express and FTP 1-2
FTP Commands Supported by Connect:Express 1-2
Access Control 1-3
Transfer Parameters 1-3
FTP Services 1-3
FTP Replies Issued by Connect:Express 1-4
Examples of FTP Sessions and Transmissions 1-4
Customizing the FTP Welcome Message 1-6
FTP Help 1-8
Using the LIST Command 1-9
Using the NLST Command 1-9
Customizing the FTP List 1-10

Connect:Express Management of FTP Transfers 1-13
Server Functions 1-13
Partner Identification 1-13
File Identification 1-13
Client Functions 1-14
FTP Transfer Request to Connect:Express 1-14
Remote Data Set Names 1-14
FTP Options 1-14
User Interface 1-15

Chapter 2 The FTP Manager

Installing the FTP Manager 2-1
The AFM Auxiliary FTP Manager 2-1
Installing the FTP Option 2-3
Creating the Start Procedure for the AFM 2-3
Defining FTP Manager Resources with the SYSPARM File 2-4
Defining the Monitor Interface with the MIFPARM File 2-5

Implementing FTP Transfers with Connect:Express 2-6
FTP Parameters in the SYSIN File 2-6

Partners Directory 2-7
Specific FTP Parameters 2-7
Managing Input Sessions 2-8
Managing Output Sessions 2-9
Files Directory 2-9
Specific FTP Parameters 2-9
Managing File Reception 2-12
Managing Output Transfer Requests 2-12
Transfer Requests 2-12
Specific FTP Parameters 2-12

Sending and Receiving Files 2-13
FTP Client Mode 2-14
FTP Server Mode 2-14
Naming the Files 2-14
Receiving the File Name from the FTP Client 2-16

FTP Protocol Options 2-17
Transfer Modes 2-17
FTP Data Types 2-17
FTP Data Structures 2-17
Using the File Structure Option 2-18
Using the Record Structure Option 2-19

Appendix A FTP Messages and Error Codes

Messages A-1
AFM Logging File A-1
Connect:Express Logging File A-2

Error Codes A-3
TRC Return Codes with FTP Transfers A-3

Appendix B Using Environment Variables

Why to Use Long Names B-1

ENVVAR Environment Variables File B-1
Syntax and Processing of the Variables B-2
Examples of Use B-3

Appendix C FTP Extended Identification

AFMFTPE File C-1
Syntax of Extended Identification C-2
How AFMFTPE File is Processed C-3
Using Exended Identification C-4
Partner Definition C-5
Transfer Request C-5

Index

Preface

The *Connect:Express z/OS FTP Guide* is for users of Connect:Express that use the FTP protocol for file transfers.

Chapter Overview

The *Connect:Express z/OS FTP Guide* is organized into the following chapters and appendices:

Chapter/Appendix	Description
Chapter 1 Connect:Express and FTP Transfers	This chapter describes Connect:Express support for the FTP transfer protocol, as well as Connect:Express options for FTP transfers.
Chapter 2 The FTP Manager	This chapter describes how to install and activate the FTP manager and how to setup FTP transfers. File naming and data processing is emphasized because they are important in managing FTP transfers.
Appendix A FTP Messages and Error Codes	This appendix describes FTP messages issued in the AFM Logging file and the specific messages and error codes issued during FTP transfers.
Appendix B Using Environment Variables	This appendix describes how to define long names for local and remote files.
Appendix C FTP Extended Identification	This appendix describes how to define long user names and passwords .

Connect:Express Documentation

Connect:Express documentation consists of the following manuals:

- The *Connect:Express z/OS 4.2.3 Release Notes* lists maintenance updates and any important notes.
- The *Connect:Express z/OS 4.2.3 Installation Guide* describes the planning and installation of Connect:Express.
- The *Connect:Express z/OS 4.2.3 User Guide* includes general information on using the TSO/ISPF interface, and serves as a reference of user and environment commands.
- The *Connect:Express z/OS 4.2.3 Utilities Guide* describes the optional Utilities package that you can integrate with Connect:Express.

- The *Connect:Express z/OS 4.2.3 FTP Guide* provides you with the information that you need to use Connect:Express with the FTP protocol.
- The *Connect:Express z/OS 4.2.3 Administration Guide* provides detailed information about transfer operations for system administrators and other advanced users of Connect:Express.
- The *Connect:Express z/OS 4.2.3 Options Guide* provides information about the CICS, IMS, and RJE interfaces available for Connect:Express.
- The *Connect:Express z/OS 4.2.3 PeSIT User Fields Guide* describes how you can exchange the PeSIT Pi37 and Pi99 fields with any PeSIT software.
- The *Connect:Express z/OS 4.2.3 Etebac3 User Guide* provides you with the information that you need to use Connect:Express with the Etebac3 protocol.
- The *Connect:Express HTTP Option Implementation Guide* provides you with the information that you need to implement HTTP access to Connect:Express z/OS repository.
- The *Connect:Express z/OS 4.2.3 SSL Guide* includes general information on implementing secured file transfers.
- The *Connect:Express z/OS 4.2.3 Sysplex Supervision Guide* includes general information on implementing a group of Connect:Express Plex managers under control of a Connect:Express Plex supervisor.

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question regarding a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at *www.sterlingcommerce.com* is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at *www.sterlingcommerce.com/customer/tech_support.html* for specific information on getting support for Sterling Commerce products.

Conventions Used in This Guide

The *Connect:Express z/OS FTP Guide* uses certain notational conventions. This section describes the conventions used in this guide.

Convention	**Description**
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.
Lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, index1.index2.PARMLIB indicates that you must provide the first and second indexes of the string. "PARMLIB" is mandatory.
Bold Letters	Bold print in syntax boxes indicates Connect:Express commands and required parameters. For example, PLEX=N indicates that the parameter PLEX must be set to N.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, PLEX=Y|N specifies that the default for PLEX is N.

Convention	Description
Vertical Bars (|)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example RUN=H|C specifies that H or C is valid.
`Monospaced characters` (characters of equal width)	`Monospaced characters` represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.
£ or #	The Pound character (£) and the hash character (#) are equivalent.

Chapter 1

Connect:Express and FTP Transfers

This chapter describes Connect:Express support for the FTP transfer protocol, as well as Connect:Express options for FTP transfers.

Introduction

Transmission Control Protocol/Internet Protocol (TCP/IP) which uses File Transfer Protocol (FTP) has become a standard. All TCP/IP products must be delivered with an FTP utility. FTP works across multiple platforms and offers wide spread use, but it lacks an application interface to provide an automated File Transfer Process.

Connect:Express provides support for FTP to meet this need. The FTP option in Connect:Express enables applications and users to exchange files in an organized and secured manner. Users can use existing FTP applications by enabling them to exchange certain Files defined in the Files directory with recognized Partners from the Partners directory. You must configure the FTP option during Connect:Express implementation.

Connect:Express focuses on file transfer FTP facilities like logging capabilities, file transmission, and file reception. FTP file management functions like renaming and deleting files, are not supported.

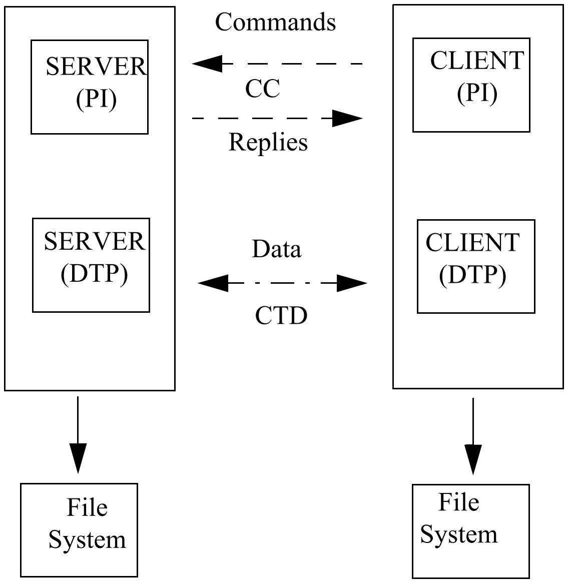
The FTP Protocol

The FTP client opens the control connection (CC) with the FTP server (Protocol Interpreter). The exchange of commands between client and server during this connection enable Connect:Express to identify the incoming FTP call or to connect to a remote FTP Server, and then negotiate one or more successive file transfers. The data transfer takes place on another CTD connection established between the server and the client Data Transfer Process (DTP).

Two modes of CTD connections are supported:

- In the default mode, the server initiates the CTD connection using the address and port number sent by the client in the PORT command. The port used for establishing the session with the remote Port is the control connection port minus one.
- In the passive mode, called PASV, the client initiates the CTD connection using the address and port number sent by the server in the REPLY 227 message.

The CC and CTD connections can be setup between two distinct sites with different TCP/IP addresses. Connect:Express supports only the unique TCP/IP address configuration. The diagram below shows the simplest example of an FTP transfer where negotiation and execution of file transfer are done between the same sites.



Connect:Express and FTP

Connect:Express supports an automated file transfer process using FTP by recognizing FTP commands and issuing replies (server mode) or sending FTP commands and recognizing replies (client mode). This section describes the FTP commands supported by Connect:Express and the replies issued by the Connect:Express FTP server.

FTP Commands Supported by Connect:Express

There are three types of FTP commands: access control commands, transfer parameters, and FTP services. The following three tables list the FTP commands for each group. The commands supported by Connect:Express are in bold.

Access Control

Command	Description	Comment
USER	Username	see “Partner Identification”
PASS	Password	see “Partner Identification”
ACCOUNT	Account	
CWD	Change Working Directory	
CDUP	Change To Parent Directory (change directory, up)	
SMNT	Structure Mount	
REIN	Reinitialize	
QUIT	Quit	

Transfer Parameters

Command	Description/option	Comment
PORT	TCP/IP address plus port number	
PASV	Passive Mode	
TYPE	**ASCII** (PRINT, **NOPRINT**) **EBCDIC** (PRINT, **NOPRINT**) **IMAGE** LOCAL byte size	default value
STRU	**File** **Record** Page (not supported)	default value
MODE	**Stream** **Block** **Compressed**	defauld value

FTP Services

The Connect:Express client side does not implement the following display services: LIST, STAT, SYST, PWD, and HELP. The commands listed below are supported by Connect:Express.

Command	Description/option	Comment
RETRieve ’pathname’	Reception by the client of the file called ’pathname’	see “File Identification”
STORe ’pathname’	Transmission by the client of the file called ’pathname’	see “File Identification”
STOU ’pathname’ STOre Unique	Transmission by the client of the file called ’pathname’	see “File Identification”
APPEnd ’pathname’		

Command	Description/option	Comment
ALLOcate		
RESTart		
RNFR	ReName From	
RNTO	ReName TO	
ABORt	Abort	
DELEte	Delete	
RMD	ReMove Directory	
MKD	MaKe Directory	
PWD	Print Working Directory	server mode
LIST	List	server mode, See Connect:Express Reply to the List command.
NLST	Name LiST	server mode. See Connect:Express Reply to the NLST command.
SITE		
SYSTem	System	server mode
STATus	Status	server mode
HELP	‘item’	server mode. See FTP replies by Connect:Express FTP Server.
NOOP		

FTP Replies Issued by Connect:Express

The Connect:Express FTP server builds replies according to standard syntax and information added by the monitor. Replies include the following information:

- Request number generated by Connect:Express (STOR, STOU, RETR).
- Results, number of bytes exchanged, and number of records processed by Connect:Express.
- Return codes TRC, SRC.

For example, FTP reply 125 is sent by Connect:Express with a request number at the end, as shown below.

```
125 STOR Command accepted, tranfer starting REQUEST : 00000018
```

Examples of FTP Sessions and Transmissions

The following screen shows an example of FTP replies sent by the Connect:Express FTP server to a UNIX FTP client. In this example, Connect:Express replies are in bold, commands and parameters from a user are underlined, and explanations are italicized.

FTP Session with Connect:Express

```
ftp> open z/OS 4100
Connected to z/OS.paris_labs.ig.stercomm.com.
220-Connect:Express For z/OS FTP monitor TOM422    (02/27/04).
220-A STERLING COMMERCE product.
220 Job: STC08854  03/01/04 16:23:58

         OPEN command is answered:
         Version of Connect:Express software  (product date)
         The system identification of the Connect:Express FTP Server Process and the
         connection date and time.

Name (z/OS:usr): ftp1
331 FTP1     password please ?
Password: _________
230-FTP1     User logged on at Connect:Express FTP z/OS.
230 Default profile is FTPV
  USER / PASW commands are returned the default profile: FTPV.
ftp> dir
200 PORT Command executed.

         DATA session is opened.

125 LIST Command accepted.
-D 2 T VB  00244 18000 FTPGDG!PSR$TST.GDG.TSTGDG0(+01)
-D 2 * VB  00244 27800 FTPV!PSR$TST.A.VVV.&REQNUMB
-F 1 R -   -     -     FTPVAL1!PSR$TST.A.VVV
250 list completed successfully.

         The LIST of symbolic files from Connect:Express Files directory available for
         User FTP1 is sent. A number of File attributes are showed. Default profile FTPV
         is part of the list. The Following attributes are sent:
         - Dynamic or Fixed Allocation
         - Allocation rule: 2 = to be created, 1 = pre-allocated, 0=to be created or
         replaced
         - Direction Transmission, Reception, * = both
         - File record format (Variable, Fixed, Blocked..)
         - Record length
         - Block size

ftp> pwd
257 "FTPV     " is current profile.
ftp>

         PWD command (Print Working Directory) is returned the default pro file.

ftp> rstat
211-Connect:Express z/OS FTP monitor TOM410    (06/27/01).
211-Default profile is FTPV
211 Mode: STREAM  ; type: ASCII ; Format: NON-PRINT; structure: FILE  .
ftp>

         RSTAT command (Remote status) is returned the active FTP transfer parameters
```

Transmission to a Connect:Express FTP Server

```
ftp> put
(local-file) TESTFTP.c
(remote-file) prod01.a.v.&reqnumb
200 PORT Command executed.
125 STOR Command accepted, transfer starting   REQUEST: 18300001.

         The Connect:Express transfer request number is returned.

226-Transfer completed successfully.
226 bytes RECV : 107222         records processed : 4028

         At end of transfer the bytse and records counts are returned.

107222 bytes sent in 147,8 secondes (0,7082 ko/s)
ftp> get
(remote-file) ftpf!prod01.a.v.ddd
(local-file) work.c
200 PORT Command executed.
451 DSN allocation error SRC=LOCA.                                    (TRC=3001)
ftp>
```

Customizing the FTP Welcome Message

When an FTP Client connects to the Connect:Express FTP manager, the following welcome message is displayed:

```
220-Connect:Express For z/OS FTP monitor TOM422 (02/27/04).
220-A STERLING COMMERCE product.
220 Job: STC08854 03/01/04 16:23:58
```

This message can be customized by changing or deleting any of the three lines using keywords in the AFM parameters. The keywords are SRVMSx, where x is a value of 1-3. The following screen shows how you can change the welcome message with these keywords.

```
SRVMSx=‘New text between cotes’
or
SRVMSx=’’ with empty text for deleting the line. (x=1,2 or 3)
```

In the example below, new text was entered for each keyword.

```
SRVMS1=’New line 1’
SRVMS2=’New line 2’
SRVMS3=’New line 3’
```

This results in the following message:

```
220 - New line 1
220 - New line 2
220 - New line 3
```

In the next example, line 2 was deleted.

```
SRVMS1='New line 1’
SRVMS2=’’
```

This results in the following message:

```
220 - New line 1
220 Job: STC08854 03/01/04 16:23:58
```

In the following example, the SRVMS1 keyword is entered incorrectly. It was entered after SRVMS2, so it is ignored.

```
SRVMS2=’New line 2’
SRVMS1=’New line 1’
```

This results in the following message:

```
220-Connect:Express For z/OS FTP monitor TOM422 (02/27/04).
220 - New line 2
220 Job: STC08854 03/01/04 16:23:58
```

FTP Help

FTP Help lists FTP commands, keywords, and special services. The following screen displays FTP help.

FTP Help

```
ftp> rhelp
214-The Connect:Express FTP commands are:
214-ABOR,*ACCT,*ALLO,*APPE,*CDUP, *CWD,*DELE, HELP, LIST, *MKD, *MODE
214- NLST, NOOP, PASS, PASV, PORT,  PWD, QUIT,*REIN,*REST, RETR, *RMD
214-*RNFR,*RNTO, SITE,*SMNT, STAT, STOR, STOU, STRU, SYST, TYPE, USER
214-
214-Commands preceded by "*" are not implemented
214-
214-TYPE can be ASCII, EBCDIC, IMAGE
214-STRU can be RECORD, FILE
214-MODE can be STREAM, BLOCK, COMPRESS
214-
214-HELP is available for:
214-*APPE, *LIST, *PWD, RETR, SITE, STOR, STOU, *TRC
214-
214-Connect:Express keywords:
214-REQUEST = nnnnnnnnn: transfer identification by monitor.
214-TRC = return code from monitor.
214-Profile = symbolic file name defined in the monitor directory.
214 Type HELP SITE command for Connect:Express special services description.
ftp> rhelp site
214-Connect:Express z/OS, special services.
214-C:X monitor identifies one transfer by a unique REQUEST number: nnnnnnnnn
214-nnnnnnnnn = Sequence number.
214-
214-Some REPLY show a TRC return code. This TRC is from C:X monitor.
214-Type HELP TRC Ftp command to get a list of some usual TRC codes.
214-
214-The name of the remote file you specify for STOR, RETR, STOU
214-Ftp commands must be structured this way:
214-{Profile}!{Dsname} where
214- Profile is a symbolic name (1 to 8 characters) known by C:X monitor.
214- A Default profile can be specified by C:X at connection (REPLY 125).
214- If {Profile} is omitted, this Default is kept
214- Dsname is a z/OS data set name (1 to 44 characters).
214- C:X monitor associates a z/OS Dsname to the Profile in its Directory.
214- If {Dsname} is omitted this Default Dsname is taken.
214- Dsname can contain C:X variables such as:
214  &REQNUMB, &REQTIM, &REQDAT, &PARTNID, &FILENAM.
ftp> rhelp stou
214-STOU {Profile}!{Dsname}:
214-Profile = symbolic identification by monitor (8 characters).
214-If omitted, default is taken.
214-Dsname = z/OS data set name (44 characters).
214-If omitted default is taken from Profile entry in monitor directory.
214-
214-If the request is accepted by monitor, the transfer is identified by:
214-REQUEST = nnnnnnnnn from monitor.
214-If the request is rejected by monitor, a TRC return code is issued.
214-Note the REQUEST number or the TRC code.
214-
214-File is created if not found.
214 If "dsname" exists, "dsname.{Request number}" is created.
ftp>
```

Using the LIST Command

An FTP client can ask the server for a list of all available files. Connect:Express replies to this command with a list of symbolic names and attributes from the Files directory. Attributes include the authorized transfer direction and the allocation of the file on the server side.

Note: The Connect:Express client feature does not send the LIST command.

Connect:Express builds the reply, according to the following rules:

- A symbolic file name can be associated with a Partner, a list of Partners, or all Partners ($$ALL$$).
- A symbolic file name can be used for transmission (STOR, STOU), reception (RETR), or both.
- Allocation rules defined for a symbolic file can affect the transfer request. For example, the physical name can be fixed in the directory, or a file can be pre-allocated.
- Transfer parameters must be related to the allocation parameters, such as format and length of the record.

The server sends the client the list of files that he can transfer, the direction of the transfers, and the allocation rules defined on the server side. The list is sent in a UNIX format and includes the following information:

- **D**ynamic or **F**ixed Allocation (1 character)
- Allocation rule: (1 character)
  - **2** = to be created
  - **1** = pre-allocated
  - **0** = created or replaced
- Direction: (1 character)
  - **T** = Transmission
  - **R** = Reception
  - * = both
- Allocation parameters (record format, record length, block size). Allocation parameters depend on the platform on which the Connect:Express server is running.
- Pathname as defined in File identification on page 1-9.

```
-D 2 T VB 00244 18000 FTPGDG!TEST.GDG.TSTGDG0(+01)
```

This list is the extended list. You can request a short list using the NLST command, or implement a user exit to change the format of the list.

Using the NLST Command

Using the NLST command, an FTP client can ask the server for a short list of available files. Connect:Express replies to this command with a list of pathnames that are available to this client.

Note: The Connect:Express client feature does not send the NLST command.

Connect:Express builds the reply, according to the following rules:

- A symbolic file name can be associated with a Partner, a list of Partners, or all Partners ($$ALL$$).

- The server sends the client the list of files that he can transfer. Only the pathname is sent for each file. The Pathname structure is defined in File Identification, page 1-9.

```
FTPGDG!TEST.GDG.TSTGDG0(+01)
```

This list is the short list. You can request an extended list with the LIST command, or implement a user exit to change the format of the list.

Customizing the FTP List

When an FTP client sends the FTP command 'LIST' or 'NLST', the AFM sends the list of Connect:Express symbolic files available to that client. The default structure of the information is defined by Connect:Express, as shown below:

```
-D 2 T VB 00244 18000 FTPTEST!TEST.&FILENAM.&REQNUMB
-D 2 R FB 00080 16000 FTPFILE!FILE.&PARTNID.&ORG
```

It is possible to build the FTP list according to the user that is requesting the information. The list is built from a model described in a source file called PARMFTPL with a record length of 128 characters. One record of this file defines the structure of the list for one client, a group of clients, or all clients. This file is declared in the JCL of the EAS address space, as shown below.

```
PARMFTPL DD       DSN=INDEX.TOMV420.PARMLIB(LISTFTP),DISP=SHR
```

If this file is not found or no record for the current user is found, the Connect:Express structure is sent. If an error occurs when processing this file, the following message is issued in the AFM LOG file and the Connect:Express structure is sent:

```
02/05/28 07:40:51 0002 AT1 ERROR PROCESSING FTP LIST, STANDARD FORMAT USED
```

The PARMFTPL File

The PARMFTPL file is used to customize FTP lists by user name. The example below shows how you could change the list structure.

```
000001 * List for all FTP Client whose name begins by USER (USER0001,USER0002....)
000002 USER* &DDN!&DSN. Direction=&DIR
000003 * List for all other FTP Clients
000004 $$ALL$$ &DSN..DIR&DIR
```

In this example, lines number 1 and 3 display the '*' (star) character in the first position indicating that the lines are comments.

Lines 2 and 4 display records. Each record describes one list record structure. All lines of the list are sent with this structure. The first field (position 1) can be an 8-character user name, a generic user name, or the keyword $$ALL$$. The $$ALL$$ keyword is used to define a default structure. You can also use a mask. For example, USER%1 selects names with any character in position 5.

Position 10 contains the description of the list record structure. You can use keywords in the format, &XXX or &XXX. with a dot at the end. The &XXX. (with a dot) format is replaced by the entire field, including blanks. The &XXX keyword is replaced by the unpadded field which uses no blank trailer. Using &XXX.. syntax (double dot) enables you to add a dot at the end of the unpadded field.

In line number 2, a generic user name is used. In this example, the &DDN keyword (with no dot) is replaced by the symbolic file name FTPTEST, with no blank trailer. The "!" character is concatenated, and the &DSN.

keyword (with a dot) is replaced by the physical file name "TEST.&FILENAM.&REQNUMB," with a blank trailer. The "Direction=" string is written and the &DIR keyword is replaced by the transfer direction "T." Based on this record in the PARMFTPL file, any USER* receives the following list:

```
FTPTEST!TEST.&FILENAM.&REQNUMB                                 Direction=T
FTPFILE!FILE.&PARTNID.&ORG                                     Direction=R
```

In the example on line 4, the keyword $$ALL$$ is used for any other FTP clients. Because there are two dots, the &DSN keyword (with no dot) is replaced by the physical file name "TEST.&FILENAM.&REQNUMB," with no blank trailer, and a dot is added. The "DIR" string is written. The &DIR keyword is replaced by the file direction "T." Based on this PARMFTPL file, all users whose name does not begin with USER receive the following list:

```
FTPTEST!TEST.&FILENAM.&REQNUMB.DIRT
FTPFILE!FILE.&PARTNID.&ORG.DIRR
```

Note: The maximum length of a line is 126 characters. The hexadecimal characters '0D0A' are automatically added to the end of the line. Lower case letters are supported.

The following table lists the keywords that you can use to customize the FTP lists.

Keyword	Description
&TIM	(8 characters): current time (HH:MM:SS)
&RDS/&RDS.	(44 characters): Remote DSN field from the directory or prepared in the hold request.
&API/&API.	(80 characters): API field prepared in the hold request.
&MBX/&MBX.	(8 characters): Mailbox name.
&NOT	(44 characters): This keyword is replaced by a file name from the application or the physical file name on the server. The application name is retrieved from the file definition in the Files directory or supplied in the hold transfer request. The API field is supplied in the hold request. The Remote DSN field is retrieved from the file definition in the Files directory or supplied in the hold transfer request. The Notes field has priority over the API field and Remote DSN field, and the API field has priority over the Remote DSN field.

Selecting the List Structure to Use

When the Connect:Express FTP server receives the FTP LIST command, it must select the list record structure in the PARMFTPL file from the FTP User name. When you log on to the Connect:Express FTP server, you specify a client name and password. Then, depending on the configuration, Connect:Express can either use that client name or a different partner name. The partner name is the symbolic name used to process the communications. Several clients can be processed with the same partner name.

The selection process gives priority to the client name, then the partner name. First, Connect:Express compares the FTP user name field to the client name, and then to the partner name. If no match is found, a default structure is used. If no PARMFTPL file is allocated, Connect:Express uses the Connect:Express default list record structure.

The following screen shows an example of the PARMFTPL file.

```
000001 *------------------------------------------------------------------
000002 *    0    01    1    2    2    3    3    4    4    5    5    6
000003 *...5...90....5....0....5....0....5....0....5....0....5....0
000004 JOHN*    &DDN..PSR$TST.&PAR..&USR..D&DAT.&REQDAT
000005 JO*      &DDN!PSR$TST.&PAR..&USR..D&DAT.&REQDAT
000006 %%%%NY   &DDN!PSR$TST.&PAR..&USR..D&DAT.&REQDAT
000007 $$ALL$$  &DDN.&PAR..&USR..D&DAT.&REQDAT (FOR ALL)
```

In the PARMFTPL file, if:

- $$ALL$$ is defined, this is the default.
- No $$ALL$$ is defined, the default is the Connect:Express default list record structure.
- One line matches the CLIENT name, it is selected.
- One or several masks match the CLIENT name, the mask for which more characters were fixed is selected. For example: CLIENT=JOHNNY, Line number 4 is for JOHN*, line number 5 is for JO*, line number 6 is for %%%%NY. Line number 4 is selected because the mask contains 4 characters from JOHNNY and the masks from lines number 5 and 6 contain only 2 characters from JOHNNY.

  If no match is found with the Client name and one line matches the Partner name, then the Partner is selected. If one or several masks match the Partner name, the mask for which more characters were fixed is selected.

Including Hold Transfer Requests in the List

In Connect:Express, the FTP server can record hold FTP transmission requests in its requests table. The FTP client can then connect and inquire about hold transmission requests available to the client.

You can include hold requests available to the current client in the list. You can also exclude the Symbolic files definitions.

A general parameter, HTTPLS=, has been added to the Connect:Express SYSIN file. Refer to Appendix D in the *Connect:Express z/OS Administration Guide* for more information about the SYSIN file. The following table provides some sample configurations for this parameter.

Sample configuration	Explanation
HTTPLST=(LIST,FCT,NORCT)	Corresponds to the previous version configuration. This is the default.
HTTPLST=(LIST,FCT,RCT)	Indicates that both symbolic file definitions and hold requests are included in the list.
HTTPLST=(LIST,NOFCT,RCT)	Indicates that the symbolic file definitions are not included in the list. Only hold requests are included.
HTTPLST=(NOLIST,NOFCT,NORCT)	can be used to never send any information in the list.

A hold request displays in the list with a parenthesis '(' in the first character, as shown in the following example:

```
------------------------------------------------------------------------------
FTPFILE!FILE.&PARTNID.&ORG Direction=R
(FTPHOLD!REPORT.R0002.D040227Direction=R
------------------------------------------------------------------------------
```

In the preceding example, FTPFILE line shows a symbolic definition available for reception and (FTPHOLD line shows a transmission request waiting for reception. Refer to the *Connect:Express HTTP Option Implementation Guide* for more information about defining mailboxes and access rights for these mailboxes.

Connect:Express Management of FTP Transfers

FTP Transfers are managed in the same way as other types of transfers in Connect:Express. Each FTP transfer is identified by a unique request number, and associated to a Partner in the Partners directory, and a File profile in the Files directory. You can monitor FTP transfers through the operator interface of the monitor using the Log file or journal. In addition, you can automate FTP transfers with the user interface of the monitor using batch programs, user exits, and user commands.

Server Functions

The monitor processes all transfer requests using the symbolic Partner name, symbolic File name, and physical file name. These names are received from the protocol commands sent by the FTP client or found in the directories.

Partner Identification

The symbolic Partner name is fixed during the connection, and the password and network address are verified.

File Identification

The physical and symbolic file names are obtained when you receive the RETR, STOR, or STOU commands.

```
                RETR ‘pathname’  -  STOR ‘pathname’  -  STOU ‘pathname’
         Connect:Express Pathname  = Symbolic name + physical name separated by “!”
```

Of the two parameters, each can be fixed by default in Connect:Express directories, which is the reason pathnames can be omitted or only partially entered. If no symbolic name is given, the default is used from the symbolic Partner definition or from Connect:Express parameters. If no physical name is given, the default is used from the symbolic File definition.

The physical file name is processed from the symbolic file definition in the directory and the physical file name part of the pathname. See *Naming the File* on page 2-13 of this book for more information.

The following table identifies four possibilities for the pathname of an FTP transfer request.

Pathname	Description
Parameter1!parameter2	“Parameter1” is considered the symbolic file name, “Parameter2” is considered a physical file name.
Parameter!	“Parameter” is considered a symbolic file name (‘!’ trailer). The physical file name is looked for in the symbolic file name definition.
Parameter	“Parameter” is considered a physical file name (no ‘!’ trailer). The symbolic file name is looked for in the Partner definition. Each Partner can have its own default file name. A general default file name must be defined in Connect:Express general parameters.

Pathname	Description
None	No pathname given: The symbolic File name is looked for in the Partner definition. Each Partner can have its own default file name. A general default file name must be defined in Connect:Express general parameters. The physical file name is looked for in the symbolic file name definition.

Client Functions

The Client function works like any other protocol. You can send and receive files with FTP servers, and FTP transfer requests can be chained in one session or performed in simultaneous sessions, depending on the number of incoming and outgoing sessions that the Partner is allocated.

The “USER” and “PASS” command parameters come from the DPCSID and DPCPSW local identification. They can be replaced by the Alias definitions, if any, in the Partner definition or the transfer request parameters.

Note: No FTP display services, such as LIST, STAT, SYST, PWD, and HELP, are supported. However, some extra parameters have been added to Connect:Express files to support FTP transfers.

FTP Transfer Request to Connect:Express

FTP options and remote file names are sent at request time, and they are FTP-specific. These parameters can be fixed in the symbolic file definition, or specified in the transfer request screen.

FTP transfer requests use a local file name (LOCAL DSN), and a remote file name (REMOTE DSN) or pathname, as parameters. You must also fix FTP transfer options such as transfer mode, data structure, and data type.

The remote DSN and FTP parameters can be set in the Files directory of the monitor, and are only used with an output request. They can be specified at the time of the FTP transfer request.

Remote Data Set Names

The remote data set name is the pathname sent to the server as the parameter of the FTP RETR, STOR, or STOU commands, as shown below.

```
RETR ‘pathname’  -  STOR ‘pathname’  -  STOU ‘pathname’
```

FTP Options

FTP options determine how a file is sent, and how it is stored on the server side, according to platform-specific guidelines. The following table identifies the FTP options in Connect:Express.

Option	Description
Type	Text file (ASCII, EBCDIC) or Binary
Structure	File (only the end of file is defined), or Record (end of record and end of file are defined)
Mode	Stream (no structure in the data flow), Block (the data flow is structured), Compressed (the data flow is structured and compressed)

Option	Description
Store Unique	When the Store Unique option is selected, the server is expected to create a new file. A version number is added, if the file already exists.

User Interface

The user interfaces of Connect:Express are the same for all protocols, and include the following:

- Operator interface
- Batch interface
- Program interface
- User exits
- User commands

Chapter 2

The FTP Manager

This chapter describes how to install and activate the FTP manager and how to setup FTP transfers. File naming and data processing is emphasized because they are important in managing FTP transfers.

Installing the FTP Manager

FTP transfers are performed by specific address spaces, the Auxiliary FTP Manager (AFM) and the FTP Effector Address Spaces (EAS). The AFM is started by the Connect:Express monitor (TOM). The FTP Effector Address Spaces are created by the AFM when an FTP transfer resource is needed.

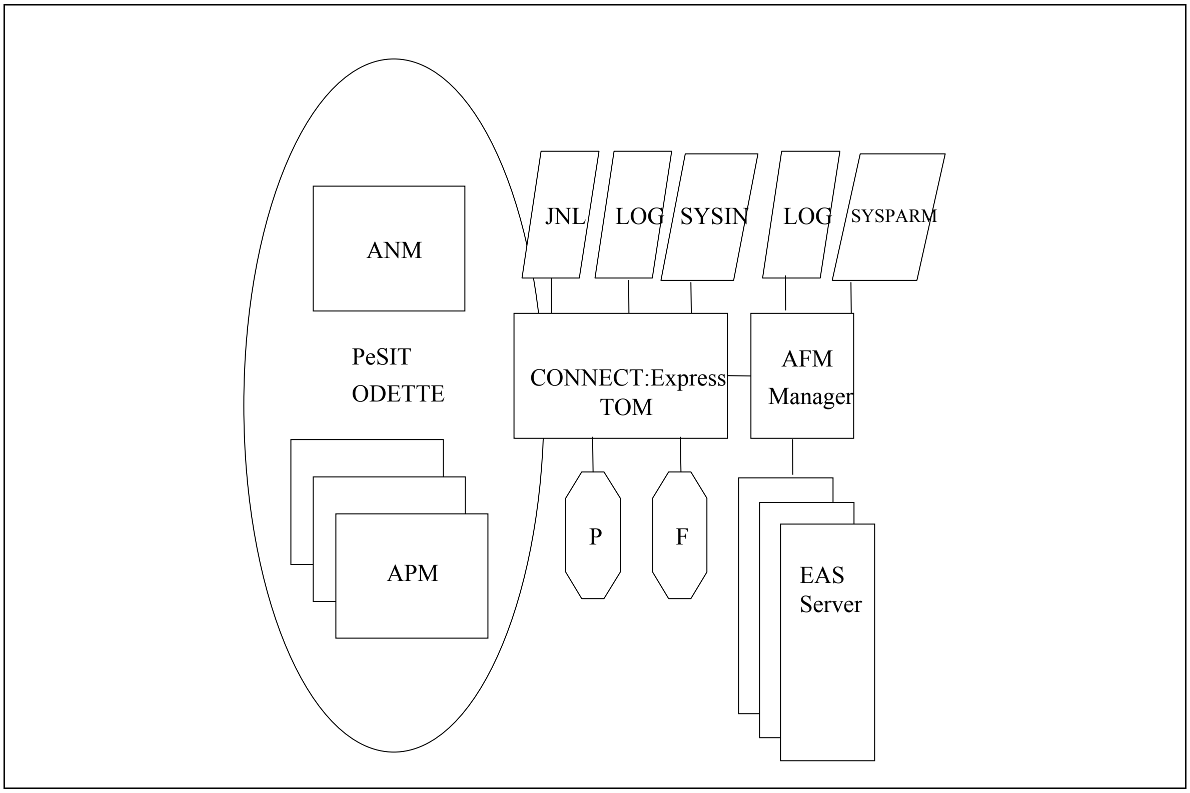
FTP transfers are managed by Connect:Express in the same way as other protocols, but FTP transfers use some specific parameters.

The AFM Auxiliary FTP Manager

The AFM has three types of processes, as described below:

1. The FTP Transfers Manager (AFM) is a unique address space started by Connect:Express. It listens to the monitor (TOM), and listens for calls on the TCP/IP port number defined in the parameters file (SYSPARM). The AFM takes transfer requests and forwards interrupts and stop commands received from Connect:Express to the active FTP transfer servers (EAS).

   This address space manages a log file in which all FTP events are registered and coded. This enables you to automate the file transfer process with FTP. The manager is in charge of accepting connections using controls from the Connect:Express Partners directory. It also transmits local FTP transfer requests to open connections with remote FTP servers. After the connection is established with a Partner, the processing is moved to the FTP transfer server (EAS) level. The following diagram shows how Connect:Express manages transfer operations for FTP transfers in the same way that it does for other protocols like PeSIT and ODETTE.



The Auxiliary FTP Manager modules communicate
with each other and with Connect:Express using cross memory services.

2. FTP transfer servers (EAS) are address spaces started by the manager after FTP establishes control of the session input or output. Each server can manage 1 to 8 simultaneous FTP sessions, and a server stays in memory as long as one FTP session is active.

   A server can remain inactive for the time specified in the SYSPARM parameters. If no new session is established within the specified time, the server stops. If the server has not reached the maximum number of parallel sessions specified in the SYSPARM parameters, the manager transmits the successive control sessions.

1. Effectors are subtasks of a server. They hold FTP control sessions and execute file transfers in FTP data sessions. An effector manages transfer negotiations under the control of Connect:Express and its Files directory. It also maintains control sessions and activates the subtasks for execution of file transfers in a data session using the following commands: STOR, STOU, RETR, LIST, and NLST.

Installing the FTP Option

Before installing the FTP option, you must install Connect:Express software. Connect:Express z/OS ships with all of the optional features, but to activate a function you must set an authorization key in the monitor’s SYSIN file. There are examples of start procedures in the *INSTLIB* library and examples of parameter files in the *PARMLIB* library. When installing the FTP option, you must:

- Copy the P0IAS000 module to LINKLIST.
- Reserve a port number for the AFM to listen for calls (input).
- Reserve the previous port number minus 1. This is used for establishing data sessions with remote FTP servers.
- Reserve a range of 255 port numbers for EAS transfer servers to monitor data sessions.
- Declare procedures started by the FTP option as TCP/IP applications. See TOMxAFM and TOMxASnn for examples.
- Declare FTP option STCs to Security as you would all other Connect:Express STCs. (TOMxAFM and TOMxAS)

After you have installed the FTP option, you must configure the FTP option and then configure Connect:Express to start the FTP server.

Creating the Start Procedure for the AFM

Connect:Express starts the AFM address space during initialization. Then, you can stop and restart the AFM using modify commands, as shown below:

- **/F TOMJOB,P AFM** requests Connect:Express to stop AFM
- **/F TOMJOB,S AFM** requests Connect:Express to start AFM

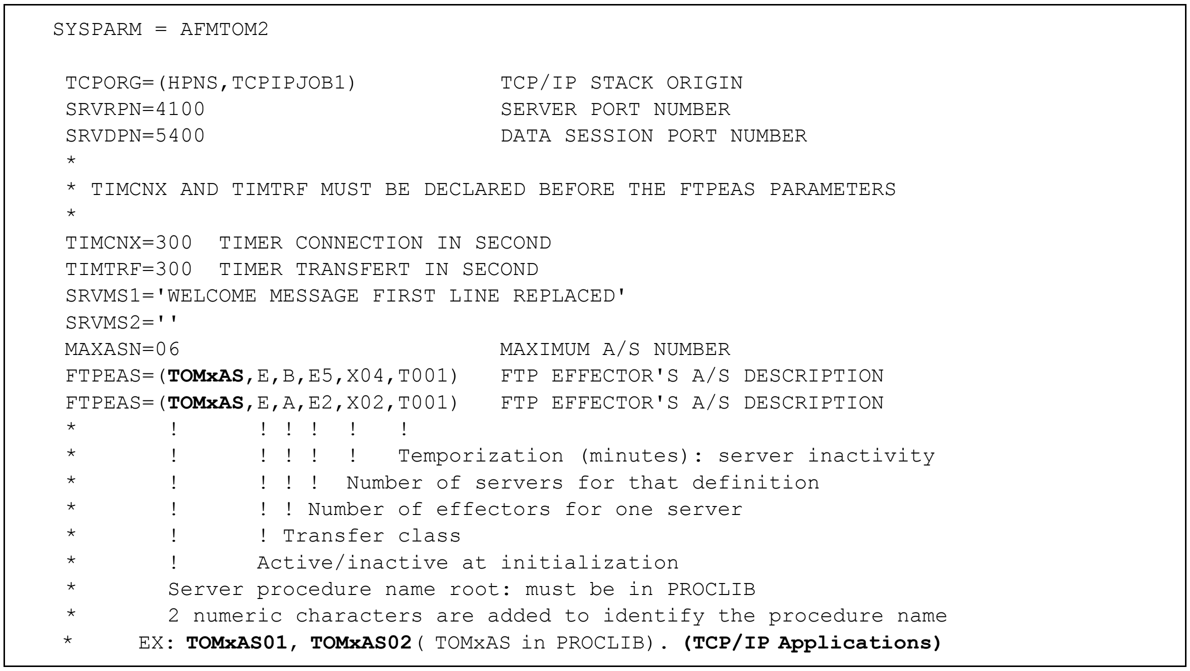
The procedure name TOMxAFM is declared in the SYSIN file of Connect:Express. See FTP Parameters in the SYSIN File on page 2-5 for more information. The following example shows the AFM procedure.

```
//TOMxAFM  PROC  SSN=                     (TCP/IP Application, RACF)
//START EXEC PGM=P1AFM000,
//         PARM=(T222,&SSN)
//STEPLIB  DD DSN=CEXPRESS.LOADLIB,DISP=SHR
//         DD DSN=YRESM80.RPTCPIP.V3R1.SEZALINK,DISP=SHR
//SYSABEND DD SYSOUT=V,HOLD=YES
//SYSLOG   DD SYSOUT=V,HOLD=YES
//AFMLOG   DD SYSOUT=V,HOLD=YES,DCB=BLKSIZE=266 LOG file of AFM
//SYSPARM  DD DSN=CEXPRESS.PARMLIB(AFM&SSN),DISP=SHR
//MIFPARM  DD DSN=CEXPRESS.PARMLIB(MIF&SSN),DISP=SHR
//AFMFTPE  DD DSN=CEXPRESS.PARMLIB(FTPEXT),DISP=SHR
//SYSCE1   DD   DSN=CEXPRESS.PARMLIB(SYSCE1),DISP=SHR
//OUTCE1   OUTPUT CLASS=U,DEST=AF3CE1
//    PEND
```

You must also consider two parameter files, the SYSPARM file where you define FTP manager resources, and the MIFPARM file where you define the monitor interface.

Defining FTP Manager Resources with the SYSPARM File

The screen below shows an example of the SYSPARM file.



```
SYSPARM = AFMTOM2

 TCPORG=(HPNS,TCPIPJOB1)           TCP/IP STACK ORIGIN
 SRVRPN=4100                       SERVER PORT NUMBER
 SRVDPN=5400                       DATA SESSION PORT NUMBER
 *
 * TIMCNX AND TIMTRF MUST BE DECLARED BEFORE THE FTPEAS PARAMETERS
 *
 TIMCNX=300  TIMER CONNECTION IN SECOND
 TIMTRF=300  TIMER TRANSFERT IN SECOND
 SRVMS1='WELCOME MESSAGE FIRST LINE REPLACED'
 SRVMS2=''
 MAXASN=06                         MAXIMUM A/S NUMBER
 FTPEAS=(TOMxAS,E,B,E5,X04,T001)   FTP EFFECTOR'S A/S DESCRIPTION
 FTPEAS=(TOMxAS,E,A,E2,X02,T001)   FTP EFFECTOR'S A/S DESCRIPTION
 *       !      ! ! !  !   !
 *       !      ! ! !  !   Temporization (minutes): server inactivity
 *       !      ! ! !  Number of servers for that definition
 *       !      ! ! Number of effectors for one server
 *       !      ! Transfer class
 *       !      Active/inactive at initialization
 *       Server procedure name root: must be in PROCLIB
 *       2 numeric characters are added to identify the procedure name
 *     EX: TOMxAS01, TOMxAS02( TOMxAS in PROCLIB). (TCP/IP Applications)
```

The following table describes the parameters in the SYSPARM File:

Parameter	Length	Description
TCPORG	(HPNS,TCPIPJOB)	The name of the TCP/IP stack. The HPNS interface is always used.
SRVRPN	4 numeric. char.	The port number on which the Connect:Express FTP server listens to incoming calls. (Port 21 is supported.) According to RFC959, the local port number used for data transfer is the session port number minus 1. For example, if the FTP server is listening on port 7001, port 7000 will be used for data transfer sessions established by the FTP server with the FTP client. This does not apply to the Passive mode (PASV) where data sessions are established by the FTP client. The next parameter (SRVDPN) is used for the passive mode.
SRVDPN	4 numeric char.	This parameter is used for Passive Mode. The base number for the port number on which the Connect:Express FTP manager is listening for a data session. The port number is computed from this parameter. Its value is in the range of SRVDPN+255.
TIMCNX	4 numeric char.	Session timer in seconds. This timer applies to the FTP connection phase, after theTCP/IP session is established and before the password is accepted.
TIMTRF	4 numeric char.	Data transfer timer in seconds. This timer applies to the data transfer phase, after the transfer request is accepted and before the end of transfer.

Parameter	Length	Description
SRVMS1 SRVMS2 SRVMS3	64 alpha char.	These optional parameters are used to customize the server logon message. The default message sent by the Connect:Express FTP server is: 220 - Connect-Express z/OS FTP monitor TOM4xx 220 - STERLING COMMERCE product . 220 - Job : STC08854 07/01/01 16 :23 :58 You can replace or delete any of the three lines. Use SRVMS1 to replace the first line, SRVMS2 to replace the second line, and SRVMS3 to replace the third line. The new string must be placed between quotes. For example, SRVMSx='THIS IS MY LINE' replaces line x. To delete a line, type an empty string, for example SRVMSx=''. These parameters must be provided in ordered list. If SRVMS1 is found after SRVMS2, SRVMS1 is omitted. Example: SRVMS1='MY MESSAGE' SRVMS2='' will result in: 220 - MY MESSAGE 220 - Job : STC08854 07/01/01 16 :23 :58 SRVMS2='' SRVMS1='MY MESSAGE' will result in: 220 - Connect-Express z/OS FTP monitor TOM4xx 220 - Job : STC08854 07/01/01 16 :23 :58 In this example, SRVMS1 is omitted because it is placed after SRVMS2.
MAXASN	2 numb. char.	The maximum number of EAS address spaces. This number must match the total FTPEAS=(…. Xnn,…) parameter.
FTPEAS	6 sub parameters separated by a comma (,): 1) 1 to 8 alphanum. char. 2) ‘E’ 3) 1 alpha. char. 4) ‘E’ + 1 numb. char. 5) ‘X’ + 2 numb. char. 6) ‘T’ + 3 numb. char.	The description of one or more transfer server address spaces. 1) The name of the procedure started by the AFM when needed. 2) Enabled at initialization. 3) Transfer class served by this server (A-Z) – future use. 4) Number of simultaneous transfers that can be performed in this address space (1 to 8). 5) Number of similar address spaces corresponding to this description. This parameter is related to the MAXASN total count. 6) Number of minutes that a client can remain inactive. When reached, the AFM cuts the ftp session. Number of minutes that this server can remain inactive. When no activity is running, the server keeps waiting for a new FTP session during this time, then the AFM shuts the server down. Example: MAXASN=04 FTPEAS=(TOM2EAS, E,B,E5,X02,T002) FTPEAS=(TOM2EAS, E,A,E3,X02,T002)

Defining the Monitor Interface with the MIFPARM File

The screen below shows an example of the MIFPARM file which you should not change.

```
MIFPARM = MIFTOM2: Connect:Express interface definition
API=SC20
APX=L0XASPB1
APN=L1B2P072
```

The following example shows the EAS procedure that is started by the AFM.

```
//TOMxAS  PROC  OUT='V,HOLD=YES',PRM=XYZ                    (TCP/IP APPLICATION,RACF)
//START       EXEC  PGM=P1AFM090,PARM=&PRM,
//           ACCT=(532,AINF)
//STEPLIB   DD   DSN=CEXPRESS.LOADLIB,DISP=SHR
//          DD   DSN=YRESM80.RPTCPIP.V3R1.SEZALINK,DISP=SHR
//SYSABEND DD   SYSOUT=&OUT
//SYSPRINT DD   SYSOUT=&OUT
//SYSUE1    DD   DSN=CEXPRESS.PARMLIB(SYSUE1),DISP=SHR
//OUTUE1    OUTPUT CLASS=U,DEST=AF3UE1
//NFYPRF    DD   DSN=CEXPRESS.NFYPRF,DISP=SHR
//SYSINFY   DD   DSN=CEXPRESS.PARMLIB(NFYPARM),DISP=SHR//*
```

Implementing FTP Transfers with Connect:Express

The FTP transfer environment is distinct from other transfer environments. To implement FTP transfers, you must define some special parameters for FTP transfers. Some of these parameters are in the SYSIN file and others are in the Partner and File directories. Monitoring and statistics for FTP transfers are the same as with other transfer types.

FTP Parameters in the SYSIN File

The FTP transfer classes definition, AFM procedure definition, FTP default profile, and the FTP list option, are general FTP parameters declared in the SYSIN file, and define the FTP resources. If they are not active, the FTP server is not active. The screen below shows an example of FTP parameters in the SYSIN file.

```
AFMCLS= (B/016/000/016) (CLAS/TOT/IN/OUT)   AFM   CLASS
AFMCLS= (A/010/008/002) (CLAS/TOT/IN/OUT)   AFM   CLASS
AFMPRC=TOMxAFM           START   AFM
AFMPRF=F1                FTP   DEFAULT PROFILE
AFMALL=N                 IF LIST $$ ALL$$ OR NOT
DPCSID=MONNAME     8     MONITOR  PARTNER NAME
DPCPSW=PASSW       8     PASSWORD
RQEMAX=0256        4     REQUEST QUEUE  ELEM.
MAXSRQ=24          2     MAX   SIMUL.    REQUEST
MSGPRX=TOM2        4     MESSAGE   PREFIX
ACTEXT=CONT        4     ACTION ONE   EXIT   ERROR
RMFLOG=Y           1     RMF   LOGGING.
...  and so on ...
```

The following table describes the FTP parameters in the SYSIN file.

Parameter	Length	Description
AFMCLS	4 subparameters separated by a slash '/' 1) 1 alpha. char. 2) 3 numb. char. 3) 3 numb. char. 4) 3 numb. char.	The description of a class of transfer profile, number of in and out FTP sessions supported. There is one card per class. Class of transfer (A-Z) Total number of simultaneous sessions Total number of incoming sessions Total number of outgoing sessions Example: AFMCLS=(A/016/008/008) AFMCLS=(B/016/000/,008) AFMCLS=(C/016/016/000)
AFMPRC	1 to 8 alphanum. char.	The name of the AFM procedure that Connect:Express starts.
AFMPRF	1 to 8 alphanum. char.	The general default profile. This symbolic file name must be defined in the file directory. This name is used if the current Partner has no default profile defined and the external FTP transfer request does not provide a symbolic file name to transfer with.
AFMALL	1 alpha. char. 'Y' , 'N' or **'S'**	This parameter is used to manage the LIST and NLIST commands received from FTP clients, and the files a client is authorized to transfer. If AFMALL=N, the response to LIST and NLIST will consist of all file definitions that are available to the current FTP Client, excluding definitions available for ALL Partners. The file definitions (screen 2/5) which has the Client name in the 'TRANSMITTING PARTNER' or 'RECEIVING PARTNER' fields will be selected. The file definition that has a Partner list in one of these two fields is selected if the client is in the list. The client is authorized to transfer $$ALL$$ files. If AFMALL=Y, the response to LIST and NLIST will consist of all file definitions that are available to the current FTP Client, including definitions available for ALL partners ($$ALL$$ or * in the Transmitting Partner or Receiving Partner fields).The client is authorized to transfer $$ALL$$ files. If AFMALL=S, the response to LIST and NLIST will consist of all file definitions that are available to the current FTP Client, excluding definitions available for ALL Partners. The file definitions (screen 2/5) which has the Client name in the 'TRANSMITTING PARTNER' or 'RECEIVING PARTNER' fields will be selected. The file definition that has a Partner list in one of these two fields is selected if the client is in the list. The client is NOT authorized to transfer $$ALL$$ files. Example: AFMALL=Y

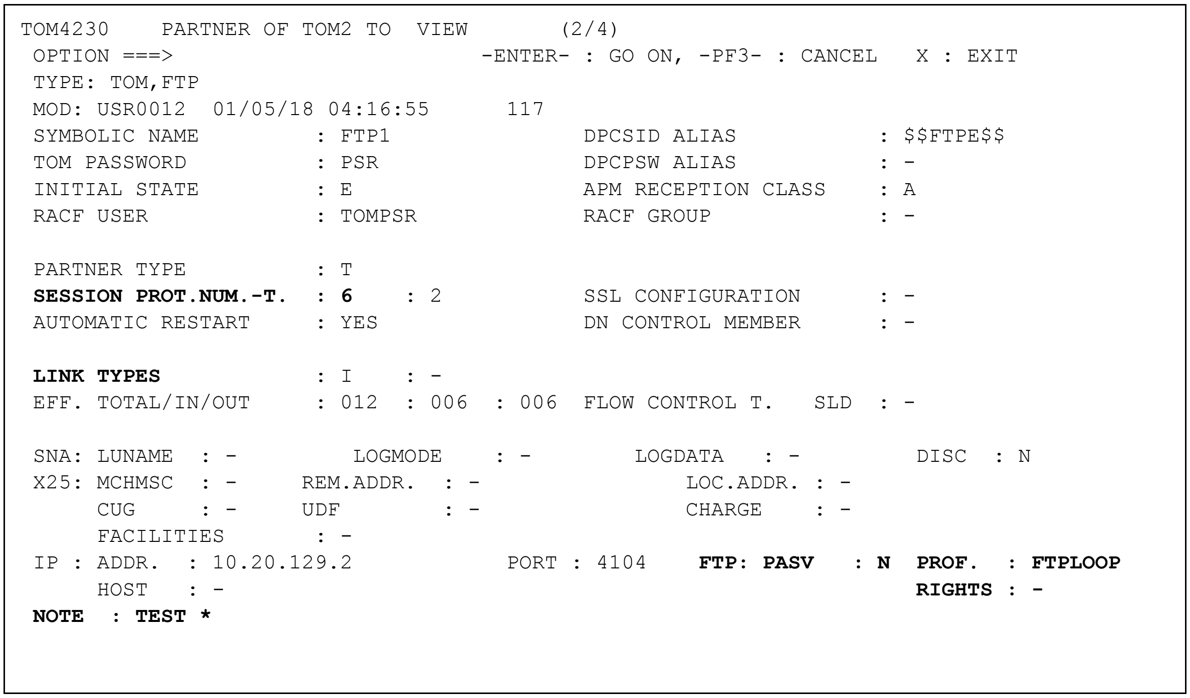
Partners Directory

FTP Partners must be defined in the Partners directory. The symbolic name of a Partner corresponds to the parameter of the FTP USER command, and the password corresponds to the parameter of the FTP PASS command.

You cannot change the transfer protocol number of an FTP partner in the Connect:Express Partners directory. If needed, you must delete the FTP Partner and create it again with the new protocol.

Specific FTP Parameters

There are FTP parameters that you must define in the Partners directory. An FTP Partner must be defined with session protocol number 6, and a link type of I. PASV and Profile are specific FTP parameters in the Partner definition. In the example below, the fields used for an FTP partner are in bold. Session Table T1B2PS6n corresponds to session protocol number 6, and has the same structure as other tables. The DPCSID ALIAS field can be used for extended identification (See *Appendix C*)



```
TOM4230    PARTNER OF TOM2 TO  VIEW       (2/4)
 OPTION ===>                        -ENTER- : GO ON, -PF3- : CANCEL   X : EXIT
 TYPE: TOM,FTP
 MOD: USR0012  01/05/18 04:16:55      117
 SYMBOLIC NAME         : FTP1                DPCSID ALIAS           : $$FTPE$$
 TOM PASSWORD          : PSR                 DPCPSW ALIAS           : -
 INITIAL STATE         : E                   APM RECEPTION CLASS    : A
 RACF USER             : TOMPSR              RACF GROUP             : -

 PARTNER TYPE          : T
 SESSION PROT.NUM.-T.  : 6    : 2            SSL CONFIGURATION      : -
 AUTOMATIC RESTART     : YES                 DN CONTROL MEMBER      : -

 LINK TYPES            : I    : -
 EFF. TOTAL/IN/OUT     : 012  : 006  : 006   FLOW CONTROL T.   SLD  : -

 SNA: LUNAME  : -         LOGMODE    : -         LOGDATA   : -         DISC  : N
 X25: MCHMSC  : -     REM.ADDR.  : -                 LOC.ADDR. : -
      CUG     : -     UDF        : -                 CHARGE    : -
      FACILITIES       : -
 IP : ADDR.  : 10.20.129.2           PORT : 4104     FTP: PASV   : N    PROF.  : FTPLOOP
      HOST   : -                                                        RIGHTS : -
 NOTE  : TEST *
```

Note: The RIGHTS field is used with the Http Option. Refer to the Http Option Guide for more information.

Managing Input Sessions

During an input connection, the FTP User and Password parameters are interpreted as symbolic name and Partner password. Connect:Express looks for the user in the Partners directory. Controls focus on the following parameters:

- Session protocol – must be ‘6’
- Password
- IP address, if any, from the directory or from the SYSTCP file
- Number of active incoming sessions for the Partner
- Status of the Partner (enabled, disabled)

Note: You can disable TCP/ IP address verification for an incoming call with the character ‘*’.

FTPBYPAS and Connection User Exit

The FTPBYPAS keyword is associated with a symbolic partner name, and can be defined in the SYSTCP file. Any calling FTP User that is not found in the Partner Directory is processed under control of this general symbolic name. If you combine FTPBYPAS with an access control user exit, the FTP User and password can be controlled in a your own User directory without defining all users in the Connect:Express Partner Directory.

M1USRCNA is a macro that enables you to easily customize the L1USRCNA exit. It is provided in the *MACLIB*. Refer to *Customizing User Access Control with M1USRCNA* in the *Connect:Express z/OS Administration Guide* for more information about this macro.

L1USRCNA is a connection server exit that can be used for security control . L1USRCNA source code is provided in the *SAMPLIB*.

Default Profile

Each Partner can be associated with a default FTP profile. This symbolic file name is defined in the Files directory. The Partner default profile is used for input connections when the symbolic file name is not specified in the FTP transfer request. The default profile from general parameters (AFMPRF in the SYSIN file) is used if it is not specified in the Partner definition.

Managing Output Sessions

An output request for a Partner invokes the connection with the remote site. Local identification parameters for the monitor, like symbolic name and password, are used as FTP user and password parameters, when no alias name is defined for the Partner. The number of active sessions for the outgoing partner is also verified.

- The PORT number is only used for an outgoing call.
- The PASV option is used for server connections through a firewall.

Note: You can define alternate TCP/IP addresses in the SYSTCP file for input and output servers.

Files Directory

Files for FTP transfer must be defined in the Files directory, and there are some specific FTP parameters that you must enter.

When receiving a file using the FTP protocol, you must provide the allocation parameters. This requirement restricts the sharing of the same symbolic file with different protocols.

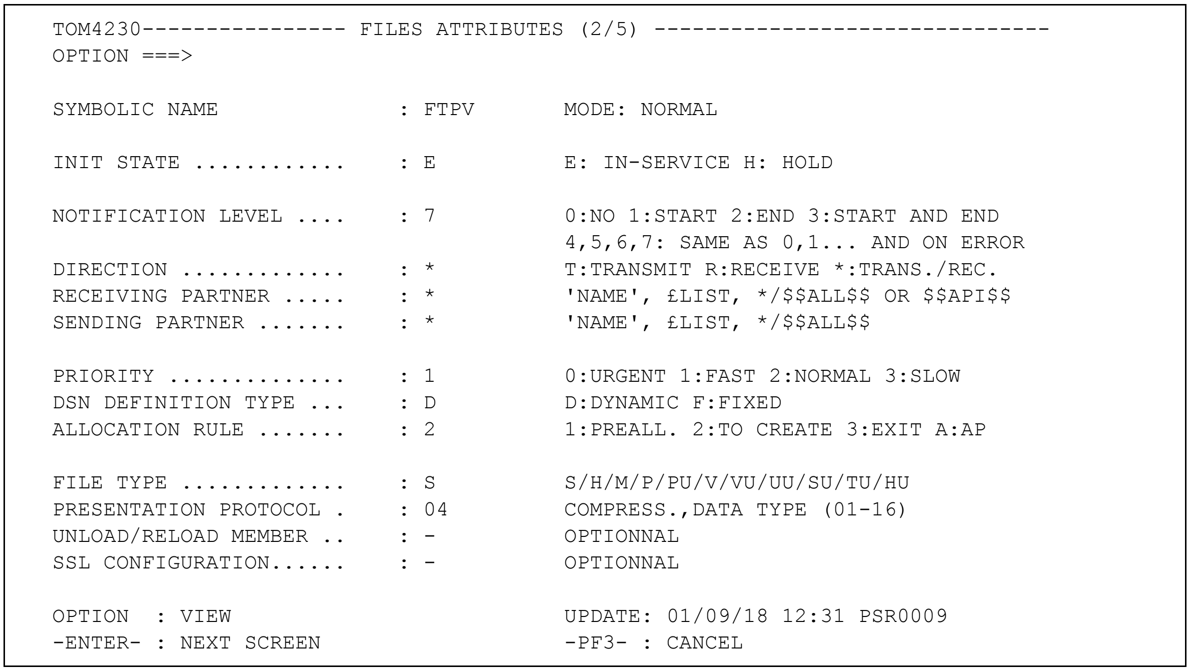
Note: The symbolic file name is used with the Store and Retrieve commands for FTP transfers.

Specific FTP Parameters

A symbolic file name in the Files directory of Connect:Express defines a transfer profile. This profile enables the management of the transfer by specifying:

- Flows for directions and Partners
- File allocation rules
- File allocation parameters for reception
- Processing rules, like exits and user commands
- The remote file name for an output request for transmission and/or reception
- FTP transfer options for an output request such as transfer mode, data structure, and data type
- Store Unique option for an output request

To send an FTP transfer request to Connect:Express, you must specify the symbolic file name. When receiving an external FTP transfer request, the symbolic file name can be provided by the FTP Client or picked up from the Partner’s default profile or the SYSIN general default profile, AFMPRF. The screen below shows normal parameters for flow of transfers and allocation rules.



```
TOM4230---------------- FILES ATTRIBUTES (2/5) -------------------------------
OPTION ===>

SYMBOLIC NAME              : FTPV       MODE: NORMAL

INIT STATE ...........     : E          E: IN-SERVICE H: HOLD

NOTIFICATION LEVEL ....    : 7          0:NO 1:START 2:END 3:START AND END
                                        4,5,6,7: SAME AS 0,1... AND ON ERROR
DIRECTION ............     : *          T:TRANSMIT R:RECEIVE *:TRANS./REC.
RECEIVING PARTNER .....    : *          'NAME', £LIST, */$$ALL$$ OR $$API$$
SENDING PARTNER .......    : *          'NAME', £LIST, */$$ALL$$

PRIORITY .............     : 1          0:URGENT 1:FAST 2:NORMAL 3:SLOW
DSN DEFINITION TYPE ...    : D          D:DYNAMIC F:FIXED
ALLOCATION RULE .......    : 2          1:PREALL. 2:TO CREATE 3:EXIT A:AP

FILE TYPE ............     : S          S/H/M/P/PU/V/VU/UU/SU/TU/HU
PRESENTATION PROTOCOL .    : 04         COMPRESS.,DATA TYPE (01-16)
UNLOAD/RELOAD MEMBER ..    : -          OPTIONNAL
SSL CONFIGURATION......    : -          OPTIONNAL

OPTION  : VIEW                          UPDATE: 01/09/18 12:31 PSR0009
-ENTER- : NEXT SCREEN                   -PF3- : CANCEL
```

Flows

You can specify the direction, Partner transmitter or receiver,and list of Partners with FTP transfers.

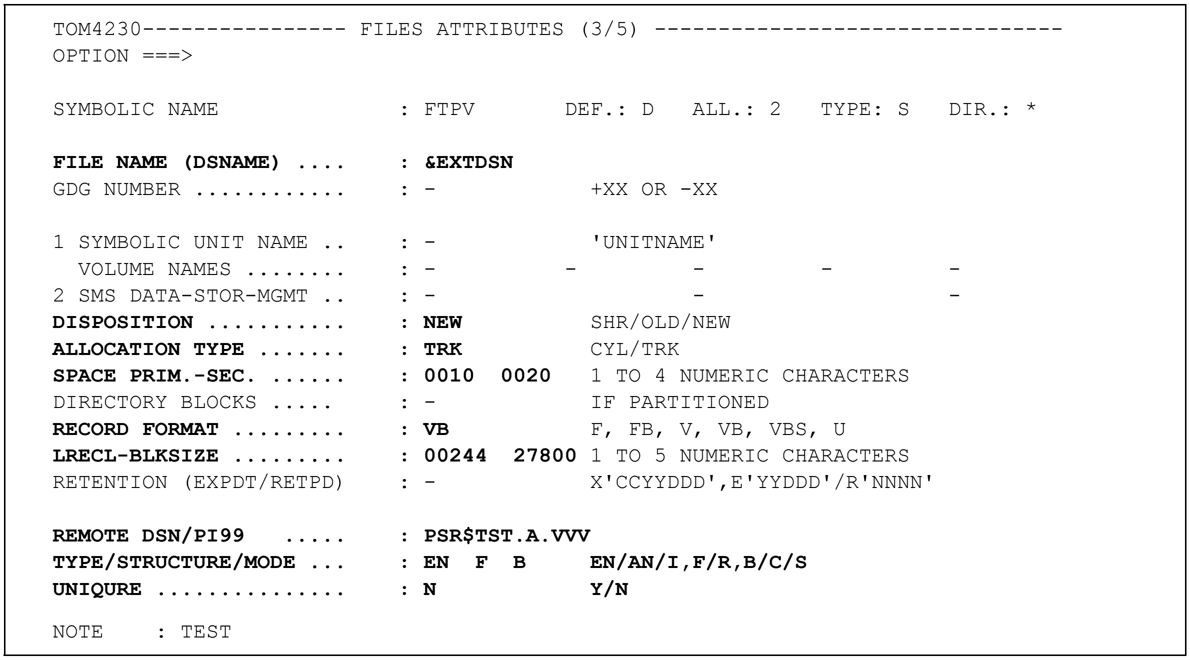
The FTP **LIST** or **NLST** command received from an FTP client invokes the transfer of the symbolic names available for transfer with the online Partner. You can also limit this list to the symbolic files for the current Partner only. The full list includes files defined with the $$ALL$$ key word in the transmitter and receiver fields. The list sent to the FTP client indicates the allocation rule and the transfer direction for each symbolic file.

Allocation Rules

Standard allocation rules apply to FTP transfers. The table below describes these allocation rules.

Rule	Description
Fixed/dynamic	Fixed means that the Files directory parameter sets the physical file name. Dynamic means that you can specify the physical file name at the time of the request.
Rule 0	The received file is replaced if it exists, or created if it doesn’t exist.
Rule 1	The file must exist. This rule is incompatible with the Store Unique option.
Rule 2	The file is created at reception time. The file should not exist unless the Store Unique command creates an index &REQNUMB when a duplicate is found.
Rule 3	The file is determined by the user exit. The file name is sent to the user exit, which you can modify and then execute.

The screen below shows the remote file name and other FTP options such as Data Type, Structure, and Mode. These fields are highlighted in bold.



```
TOM4230---------------- FILES ATTRIBUTES (3/5) --------------------------------
OPTION ===>

SYMBOLIC NAME              : FTPV       DEF.: D   ALL.: 2   TYPE: S   DIR.: *

FILE NAME (DSNAME) ....    : &EXTDSN
GDG NUMBER ...........     : -             +XX OR -XX

1 SYMBOLIC UNIT NAME ..    : -             'UNITNAME'
  VOLUME NAMES ........    : -          -        -         -         -
2 SMS DATA-STOR-MGMT ..    : -                   -                   -
DISPOSITION ..........     : NEW           SHR/OLD/NEW
ALLOCATION TYPE .......    : TRK           CYL/TRK
SPACE PRIM.-SEC. ......    : 0010  0020    1 TO 4 NUMERIC CHARACTERS
DIRECTORY BLOCKS .....     : -             IF PARTITIONED
RECORD FORMAT .........    : VB            F, FB, V, VB, VBS, U
LRECL-BLKSIZE .........    : 00244  27800 1 TO 5 NUMERIC CHARACTERS
RETENTION (EXPDT/RETPD)    : -             X'CCYYDDD',E'YYDDD'/R'NNNN'

REMOTE DSN/PI99   .....    : PSR$TST.A.VVV
TYPE/STRUCTURE/MODE ...    : EN  F  B      EN/AN/I,F/R,B/C/S
UNIQURE ...............    : N             Y/N

NOTE    : TEST
```

Remote File Name

With an output request in client mode, you can specify the pathname carried by the FTP transfer command. It can be fixed in the Files directory or specified at the time of the FTP transfer request . This field can contain symbolic and environment variables (See Appendix B).

Note: The online help provides a list of available symbolic variables. Press <PF1> and type VS in the Option field to display this list.

FTP Options

For an output request in client mode, the FTP transfer parameters can be fixed in the Files directory or specified at the time of the transfer request. These transfer parameters are listed below and described in detail on page 2-15.

- Data type: ASCII (default), EBCDIC or Binary
- Data structure: File (default) or Record
- Transfer mode: Stream (default), Block, Compress

Store Unique Option

The default value is N. When this field is set to Y, the server creates a new file whether it exists or not, and the server manages the versions of the files.

Allocation rule 1 states that when receiving a file, if it exists, it is replaced by the incoming file. This contradicts the Store Unique (STOU) command which enables a duplicate file to be written. A Store Unique command received for a file with allocation rule 1 is rejected with a TRC=2018.

Managing File Reception

Allocation parameters are mandatory if the allocation rule is 0 or 2 because no allocation parameters are provided by the FTP protocol.

Managing Output Transfer Requests

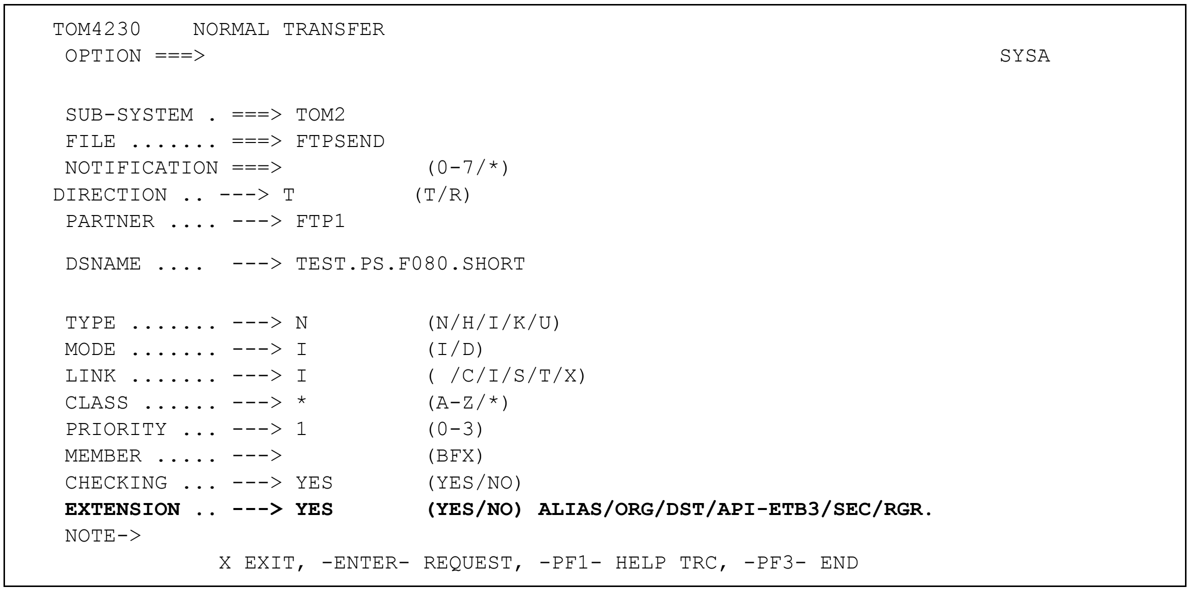
The remote file name and FTP option can be fixed in the Files directory or sent in the parameters of the transfer request. The Store Unique option can be fixed in the Files directory or sent in the parameters of the transfer request.

Transfer Requests

FTP transfer requests are made in the same way as transfer requests for other protocols. FTP transfer requests have specific FTP parameters that can be changed at the time of the request. If you do not change them, default values from the Connect:Express directories are used.

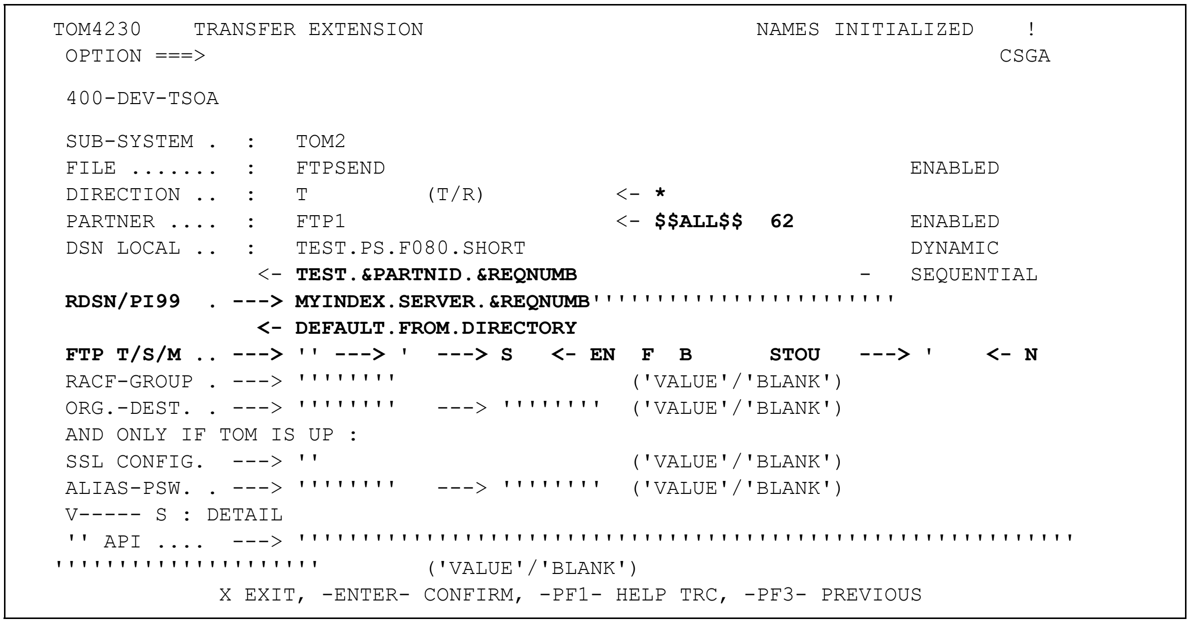
Specific FTP Parameters

The FTP transfer options, Store Unique option, and remote file name can be changed at the time of the transfer request and they replace the corresponding directory parameters. Otherwise Connect:Express uses the default parameters from the symbolic file definition. The Normal Transfer screen, shown below, enables you to define the transfer request parameters.



```
TOM4230    NORMAL TRANSFER
 OPTION ===>                                                      SYSA

 SUB-SYSTEM . ===> TOM2
 FILE ....... ===> FTPSEND
 NOTIFICATION ===>          (0-7/*)
DIRECTION .. ---> T         (T/R)
 PARTNER .... ---> FTP1

 DSNAME ....  ---> TEST.PS.F080.SHORT

 TYPE ....... ---> N         (N/H/I/K/U)
 MODE ....... ---> I         (I/D)
 LINK ....... ---> I         ( /C/I/S/T/X)
 CLASS ...... ---> *         (A-Z/*)
 PRIORITY ... ---> 1         (0-3)
 MEMBER ..... --->           (BFX)
 CHECKING ... ---> YES       (YES/NO)
 EXTENSION .. ---> YES       (YES/NO) ALIAS/ORG/DST/API-ETB3/SEC/RGR.
 NOTE->
             X EXIT, -ENTER- REQUEST, -PF1- HELP TRC, -PF3- END
```

If you select YES in the Extension option, Connect:Express displays the Transfer Extension screen. This screen enables you to define specific FTP parameters, and shows the default parameters defined in the directories.



```
TOM4230    TRANSFER EXTENSION                         NAMES INITIALIZED    !
 OPTION ===>                                                              CSGA

 400-DEV-TSOA

 SUB-SYSTEM .  :   TOM2
 FILE .......  :   FTPSEND                                            ENABLED
 DIRECTION ..  :   T          (T/R)          <- *
 PARTNER ....  :   FTP1                      <- $$ALL$$  62           ENABLED
 DSN LOCAL ..  :   TEST.PS.F080.SHORT                                 DYNAMIC
              <- TEST.&PARTNID.&REQNUMB                           -   SEQUENTIAL
 RDSN/PI99  . ---> MYINDEX.SERVER.&REQNUMB''''''''''''''''''''''''
              <- DEFAULT.FROM.DIRECTORY
 FTP T/S/M .. ---> '' ---> '  ---> S   <- EN  F  B      STOU   ---> '    <- N
 RACF-GROUP . ---> ''''''''                ('VALUE'/'BLANK')
 ORG.-DEST. . ---> ''''''''   ---> ''''''''  ('VALUE'/'BLANK')
 AND ONLY IF TOM IS UP :
 SSL CONFIG.  ---> ''                      ('VALUE'/'BLANK')
 ALIAS-PSW. . ---> ''''''''   ---> ''''''''  ('VALUE'/'BLANK')
 V----- S : DETAIL
 '' API ....  ---> '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''        ('VALUE'/'BLANK')
             X EXIT, -ENTER- CONFIRM, -PF1- HELP TRC, -PF3- PREVIOUS
```

In the example above, the file FTPSEND can be transmitted in both directions, to all Partners, and with session table number 62. The default local DSN in the directory is used for reception (TEST.&PARTNID.&REQNUMB). A default remote DSN is given in the directory and replaced with the request parameter. This field can contain symbolic and environment variables (See Appendix B).)

Note: The online help provides a list of available symbolic variables. Press <PF1> and type VS in the Option field to display this list.

The ALIAS field can be used for extended identification (See *Appendix C.* The default FTP options are ‘EN’ ‘F’ ‘B’ (EBCDIC NON PRINT, FILE, BLOCK). You can change any of these FTP options on the request screen. Stream is defined in this example, and the Store Unique option is set to ‘N’ in the directory.

The following FTP commands are sent to the FTP server after establishing the connection with FTP1.

```
TYPE EN
STRU F
MODE S
STOR MYINDEX.SERVER.&REQNUMB
```

Sending and Receiving Files

The FTP transfer request that moves from the FTP client to the FTP server carries one parameter called the “pathname.” This string is used by Connect:Express to identify both the symbolic file and the physical file to transfer. The following file naming conventions are used:

Symb – designates a symbolic file name.

DSN – designates a physical file name on the host Connect:Express-FTP (local DSN, remote DSN).

file – designates the physical name of the file exchanged.

An FTP transfer request can be performed in any of the following ways:

- STOR (Store) – received by the server.
- RETR (Retrieve)– transmitted by the server.
- STOU (Store Unique) – received by the server with a version number attribute if it is a duplicate.

Each request provides a parameter called the pathname that indicates the file name on the server side. With Connect:Express FTP, the pathname is composed of both a Symbolic name and a Physical file name separated by a !, as shown below.

```
   Pathname = ‘Symb’!’file’    or    ‘symb’!    or    ‘file’   or   ommitted
```

You can specify an inquiry with the retreive request. Place an open parenthesis , at the beginning of the pathname, as shown below, to indicate that you are inquiring about a hold transmission request in the server requests table.

```
   Pathname = (‘symb’!         (No 'file' parameter can be specified)
```

FTP Client Mode

The local symbolic file name and physical file name are the standard parameters of a transfer request in Connect:Express. The symbolic File name is mandatory and the corresponding definition in the directory includes some defaults. The physical file name can be defined in the symbolic File definition. If the DSN Definition Type is not fixed, the physical file name can be sent any time during the transfer request.

You can use the Remote DSN as a default parameter in the symbolic File definition and the actual parameter at transfer request time. The physical file name that is sent in the FTP command must match the specifications of the remote FTP server. If the remote server is a Connect:Express server, then the Remote DSN must satisfy the “symb!file” syntax.

Note: Only keywords that do not refer to the ‘file’ string, like &PARTNID, &FILENAM, or &REQ, are available in the ‘file’ string. &USRVAR1, &USRVAR2, and &EXTDSN cannot be used in the file string.

FTP Server Mode

The monitor determines the local physical name of the file to be transferred from the pathname of the FTP request. The physical file name is controlled by the Files directory parameters, and is built from the ‘LOCAL DSN’ found in the Files directory (entry: SYMB). The typical keywords are available (&PARTNID, &FILENAM, &REQ..., &USRVAR1, &USRVAR2, &EXTDSN).

When &USRVAR1, &USRVAR2, and &EXTDSN are used, the physical file name from the client helps build the local physical name.

Note: The file issued from an FTP Client can be any character string, and the Connect:Express FTP server changes it to make it available on the local platform.

Naming the Files

There are two ways to name a data set when a Partner requests a transfer. Dynamic keywords enable you to select one of the two methods.

- You can attach a local definition to the symbolic file name in the directory. In this case, the FTP client does not know about the server naming conventions.

- ❖ You can use the remote data set name sent by the FTP Client. In this case, the Client manages the names of files on the server site.

Using the Local Data Set Name

You can declare the physical name of the file to be sent or received on the file definition screens, and introduce keywords that make this name dynamic. If you use any dynamic keywords except &EXTDSN, &USRVAR1 and &USRVAR2, the physical name sent by the client in the pathname is not considered.

In the example below, the name of the Partner is USER1, and the symbolic name of the file is SYMB.

Pathname Received	Local Data Set Name Definition	Data Set Name Processed by the Server
SYMB!data.set.name	Index1.index2.index3	Index1.index2.index3
SYMB!data.set.&PARTNID	Index1.&FILENAM.&REQNUMB	Index1.SYMB.A0000115

Using the Remote Data Set Name

You can declare the physical name of the file to be sent or received on the symbolic file definition screens. The following keywords make the Connect:Express FTP Server process the physical name from the pathname received from the FTP client:

Keyword	Description
&EXTDSN	The full physical name is taken from the pathname.
&USRVAR1	The last index of the physical name from the pathname is taken.
&USRVAR2	The before last index of the physical name from the pathname is taken.

A combination of fixed indexes and any dynamic keywords can be used in both physical names, the local definition, and the pathname. In the following example, the Partner name is USER1, the symbolic file name is SYMB.

Pathname Received†	Local Data Set Name Definition	Data Set Name Processed by the Server
SYMB**!**data.set.name	Index1.&EXTDSN	Index1.data.set.name
SYMB**!**data.set.name	Index1.&PARTNID.&EXTDSN	Index1.USER1.data.set.name
SYMB**!**data set.&PARTNID	Index1.&USRVAR1.index2	Index1.USER1.index2
SYMB**!**dataset.&PARTNID	Index1.&USRVAR1.&REQNUMB	Index1.USER1.A0000115

† &EXTDSN and &USRVAR1 & 2 are not allowed in the pathname.

Using the Store Unique Option

The Store Unique option is used to ensure that the server accepts the file, when there is a chance of a duplicate file. When receiving the ‘STOU’ FTP command, the Connect:Express FTP server verifies the allocation rule, then the data set name is computed using the conventions described in the next section. If the data set name is found on the disk, an index is added at the end. This index is built from the request number.

Note: You cannot use allocation rule = 1. If you do, you will get a TRC 2018.

In the example below, the name of the Partner is USER1, and the symbolic name of the file is SYMB.

STOU Pathname Received	Local Data Set Name Definition	Data Set Name Processed by the Server
SYMB!data.set	Index1.index2.index3	Index1.index2.index3 Duplicate: Index1.index2.index3.A0000236
SYMB!data.&PARTNID	Index1.&USRVAR1.&REQNUMB	Index1.USER1.A0000115 Duplicate: Index1. USER1.A0000115.A0000115

Receiving the File Name from the FTP Client

You can receive an FTP command from a remote system, as shown below.

```
   STOR FTEST!index1:index2\&partnid/file-name/application/module-name-1
```

In this example, 'symb' and 'file' are given.

- The symbolic file is **FTEST.**
- The file string is more than 44 characters.
- Separators are not '.'.
- The third index contains a keyword.

The Connect: Express FTP server processes the command above using the FTEST definition in the directory:

```
   FILE NAME (DSN) = &EXTDSN
```

In this example:

- Connect:Express puts all characters in upper case.
- The 44 physical file name last characters received are taken into account:
  (INDEX1:INDEX2\)&PARTNID/FILE-NAME/APPLICATION/MODULE-NAME-1
- Invalid characters for a data set name on the z/OS system are compacted and replaced by points.
  &PARTNID.FILE-NAME.APPLICATION.MODULE-NAME-1
- Indexes are truncated to 8 characters.
  &PARTNID.FILE-NAM.APPLICAT.MODULE-N
- The variable &PARTNID is replaced.
  Resulting data set name is = PARTFTP.FILE-NAM.APPLICAT.MODULE-N

A second example is shown below.

```
   FILE NAME (DSN) = RACINE1.RACINE2.&PARTNID.&REQNUMB
```

The 'file' string received is ignored (no &EXTDSN, &USRVARi keyword), and the resulting data set name is = RACINE1.RACINE2.PARTFTP.A1700004

FTP Protocol Options

FTP options like TYPE, STRUcture and MODE affect how data is processed. The Type option indicates if the file is a text file or a binary file, the Structure option indicates if the end of record and end of file are indicated in the data flow, and the Mode option indicates if the data is sent as a stream or segmented into blocks.

The FTP options and the z/OS record format of the file are combined to store data in different ways. Both null length records and variable length records can be kept, deleted, or filled in with blanks. The following rules apply:

- To work as other transfer protocols, the translation is done by a user exit declared in a presentation table, with no consideration of the data type.
- The file structure allocated for reception must enable the storing of data and keep their structure intact when it is required.
- Padding is activated for files with fixed RECFM (F, FB….), and a null length record is replaced by a blank record.
- Null records are deleted for files with undefined RECFM (U).
- No un-padding is done during transmission.

The variable format enables you to keep the data structure with no change (no padding, null record kept). The two data storage tables show the possible combinations of data type and data structure options.

Transfer Modes

The transfer mode (Stream, Block, Compress) does not influence the way data is stored.

FTP Data Types

Three types of data are supported:

Data Type	Description
ASCII	Text files structured by separators, possibly completed by blanks before storing on z/OS; separator X‘0D0A’ is deleted, and the LRECL of the file must be correct. When sending text files, Connect:Express adds the X‘0D0A’ separator. No unpadding is done.
EBCDIC	Text files structured by separators, possibly completed by blanks before storing on z/OS; separator X‘15’ is deleted, and the LRECL of the file must be correct. When sending text files, Connect:Express adds the X‘15’ separator. No unpadding is done.
BINARY	Data is not submitted for any processing before storing.

FTP Data Structures

Two data structures are supported, as described in the table below.

Data Structure	Description
FILE	The record separator, if any, is part of the data. Binary type data, with no separator, is segmented and stored in the file.

Data Structure	Description
RECORD	The record separator is part of the protocol information. Binary type data is stored by keeping the transportation record size (determined by the transmitter software).

The processing is different for binary type data because ASCII and EBCDIC text files have delimiters separating each record.

Using the File Structure Option

The following table shows how data transferred with the File structure can be stored according to the record format. The correct record length must be allocated. The written file can be different from the original file (padding, null records deleted).

Transfer Definition	Fixed Record Format	Variable Record Format	Undefined RecordFormat
SFAN Stream File ASCII Non print BFAN Block File ASCII Non print.	LRECL must be long enough. **PADDING is** enabled. Null record changed to Blank record. Padding is kept if retransmitted.	LRECL must be long enough. No padding. Null record is kept. File stored is identical to original file.	LRECL must be long enough. No padding. Null record is deleted. Null records are lost if retransmitted.
SFEN Stream File EBCDIC Non print BFEN Block File EBCDIC Non print	LRECL must be long enough. **PADDING is** enabled. Null record changed to Blank record. Padding kept if retransmitted.	LRECL must be long enough. No padding. Null record is kept. File is identical if retransmitted.	LRECL must be long enough. No padding. Null record is deleted. Null records are lost if retransmitted.
SFB Stream File Binary BFB Block File Binary	LRECL is indifferent. Data segmented: length of segment =LRECL. File length must be multiple of LRECL. File is identical if retransmitted.	LRECL is indifferent. Data segmented: length of segment=LRECL-4. Last segment may be short. File is identical if retransmitted.	LRECL is indifferent. Data segmented: length of segment=LRECL. Last segment may be short. File is identical if retransmitted.

Using the Record Structure Option

The following table shows how data transferred with the Record structure can be stored according to the record format The correct record length must be allocated.

The written file can be different from the original file (padding, suppression of null records). This table is just like the previous table, except the information in the third row for binary data types is different. The record is written as it is received, with no segmentation, and the allocated LRECL must fit the record length of the data received.

Transfer Definition	Fixed Record Format	Variable Record Format	Undefined Record Format
SFAN Stream File ASCII Non print	LRECL long enough. **PADDING is** enabled. Null record changed to Blank record.	LRECL long enough. No padding. Null record is kept.	LRECL long enough. No padding. Null record is deleted.
BFAN Block File ASCII Non print	Padding is kept if retransmitted.	File stored is identical to original file.	Null records are lost if retransmitted.
SFEN Stream File EBCDIC Non print	LRECL long enough. **PADDING is** enabled. Null record changed to Blank record.	LRECL long enough. No padding. Null record is kept.	LRECL long enough. No padding. Null record is deleted.
BFEN Block File EBCDIC Non print	Padding is kept if retransmitted.	File is identical if retransmitted.	Null records are lost if retransmitted.
SFB Stream File Binary BFB Block File Binary	LRECL must be >= to the length of the record received. The size of the record can correspond to the data message length or to the file structure on the sender site. When sent by Connect:Express, the size of the record is equal to the data message length.	LRECL-4 must be >= to the length of the record received. The size of the record can correspond to the data message length or to the file structure on the sender site. When sent by Connect:Express, the size of the record is equal to the data message length.	LRECL must be >= to the length of the record received. The size of the record can correspond to the data message length or to the file structure on the sender site. When sent by Connect:Express, the size of the record is equal to the data message length.

Appendix A

FTP Messages and Error Codes

This appendix describes FTP messages issued in the AFM Logging file and the specific messages and error codes issued during FTP transfers.

Messages

The AFM address space manages its own logging file. FTP specifics appear in the Connect:Express logging file.

AFM Logging File

The AFM logging file provides information such as parameters exchanged between the FTP partners, TCP/IP network return codes, and allocation errors. This information is related to the Connect:Express logging file information. The screen below shows an example of the AFM logging file.

```
1                    2  3      4        5      6   7        8   9     10   11     12     13        14     15
01/07/03 11:40:14 0000 IN1 REJECTED     FTP1    I                                                   00404  W4 P
01/07/03 11:40:34 0001 IN1 REJECTED     FTP1    I                                                   00404  W5  J
01/07/03 11:40:52 0000 IN1 LOGGED ON    FTP1    I                                                   00404  W5
01/07/03 11:41:25 0000 IT1 ACCEPTED     FTP1    I  FTPV    STOR SFAN P:125 T:0000 S:0000 U:00000000 00408  W5  DQ
01/07/03 11:41:25 0000 CT1 $TST.A.VVV.A0000019                     FTP-00000019                     00408  W5   Q
01/07/03 11:42:48 0001 IN1 LOGGED ON    FTP1    I                                                   00404  W5  )è
01/07/03 11:43:59 0000 IT3 ENDED        FTP1    I  FTPV    STOR SFAN P:226 T:0000 S:0000 U:00000000 00408  W5L )Y
01/07/03 11:44:00 0001 IT1 ACCEPTED     FTP1    I  FTPV    RETR SFAN P:125 T:0000 S:0000 U:00000000 00408  W5M tH
01/07/03 11:44:00 0001 CT1 TST.A.VVV                               FTP-00000019                     00408  W5M
01/07/03 11:44:00 0001 IT3 ENDED        FTP1    I  FTPV    RETR SFAN P:226 T:0000 S:0000 U:00000000 00408  W5M=
01/07/03 11:46:24 0002 IN1 LOGGED ON    FTP2    I                                                   00404  W6;
01/07/03 11:46:42 0002 IT1 ACCEPTED     FTP2    I  FTPOS390RETR SFAN P:125 T:0000 S:0000 U:00000000 00408  W6>  F
```

The following table describes the fields in the AFM logging file.

Field	Description
1	Date and hour of message in the LOG file of AFM.
2	Session Number attributed by AFM.
3	Definition of the type of message: **xyn** **x** = type: Information, Anomaly, Complement, Warning **y** = nature: FTP Negotiation, FTP control Session, FTP Data Session, Data Transfer, AFM services **n** = stage: 0 initialization, 1 initialized, 2 running, 3 termination, 4 ended (OK or NOK), 5 aborted
4	Libel: ACCEPTED, REJECTED, ENDED, LOGGED ONE, INTERRUPTED.

Field	Description
5	Symbolic partner name
6	Direction of session: **I**nput, **O**utput
7	Symbolic file name
8	FTP Command: STOR, STOU, RETR
9	FTP Transfer Option: **abcd** **a** = transfer mode: **S**tream, **B**lock, **C**ompress **b** = data structure: **F**ile, **R**ecord **c** = data type: **A**scii, **E**bcdic, **B**inary **d** = **N**on print
10	Protocol return code: PRC = FTP REPLY NUMBER
11	Connect:Express Return code: TRC
12	System Return code: SRC
13	User exit Return code
14	Address space identification (The AFM or server issued the message.)
15	Time and date in binary format (TOD CLOCK)
16	Physical data set name (Information message)
17	Request Number from Connect:Express (Information message)

The following message is issued in case of outgoing connection error:

```
01/07/03 11:40:34 0001 002 CONNECTION    FTP1               O          N:0000003D
01/07/03 11:40:34 0001 002 TCPIP ERROR   FTP-00000001                  0000003D
```

In the example above, the return code 0000003D must be converted to a decimal value and searched for in the IBM documentation: z/OS V2R7.0 eNetwork CS IP and SNA Codes. If the return code decimal value is less than 1000, see section 12.1 System Error Return Codes. If the return code is more than 1000, see section 13.1 Sockets Extended Return Codes.

Connect:Express Logging File

This specific FTP information is found in the session messages in the PRC fields and in the APM fields (APM 11 identifies the AFM). All other messages are identical to any other protocol information.

```
01/07/03 15:21:30 AFM INCOMING FTP SESSION (11/01) OPENED     WITH FTP1(192.020.020.005)
01/07/03 15:21:52 REQUEST 00000025 FTPV     TRANSFER ACCEPTED APM 11  EFF 01
01/07/03 15:21:55 REQUEST 00000025 FTPV    TRANSFER ABORTED  TRC=3001 PRC=0451  SRC=010C         APM 11  EFF 01
01/07/03 15:22:34 AFM INCOMING FTP SESSION (11/01) CLOSED     WITH FTP1     (192.020.020.005)   TRC=0000
01/07/03 15:29:42 AFM INCOMING FTP SESSION (11/02) OPENED     WITH FTP1     (192.020.020.005)
01/07/03 15:30:02 REQUEST 00000026 FTPV     TRANSFER ACCEPTED                                   APM 11  EFF 02
01/07/03 15:30:11 REQUEST 00000026 FTPV     TRANSFER STARTED                                    APM 11  EFF 02
01/07/03 15:30:12 REQUEST 00000026 FTPV     TRANSFER ENDED                                      APM 11  EFF 02
01/07/03 15:30:12 REQUEST 00000026  RECEIVED     <- FTP1    ,   FILE  FTPV   NUMBER OF RECORDS: 000000000010
01/07/03 15:30:12 TRANSFER DURATION  : 00:00:08 , RESTART NUMBER     : 000 , NUMBER OF S/R    :     00000002
01/07/03 15:30:12 REQUEST 00000026 <- FTP1      COMPLETED
01/07/03 15:30:12 REQUEST 00000026 <- FTP1      PURGED
01/07/03 15:30:42 AFM  INCOMING FTP SESSION (11/02) CLOSED    WITH FTP1     (192.020.020.005)   TRC=0000
01/07/03 15:33:17 AFM  INCOMING FTP SESSION (11/01) OPENED    WITH FTP1     (192.020.020.005)
01/07/03 15:33:34 REQUEST 00000027 FTPV     TRANSFER ACCEPTED                                   APM 11  EFF 01
01/07/03 15:33:35 REQUEST 00000027 FTPV     TRANSFER ABORTED  TRC=3001 PRC=0451SRC=LOCA         APM 11  EFF 01
```

Error Codes

Standard Connect:Express return codes (TRC) are combined with specific FTP protocol return codes (PRC). PRC are the FTP reply numbers. For example, PRC=0553 means that reply number 553 has been sent or received.

TRC Return Codes with FTP Transfers

When a transfer is rejected by Connect:Express, return codes are issued. The following table lists the type of errors that can occur.

Error Type	Code	Description/Result
FTP negotiation error	**TRC=15**.. (invalid parameter)	In these cases, the request is not transmitted to the monitor
	TRC=16.. (command rejected)	
	TRC=11.. (invalid command length)	
User identification, transfer request rejected by monitor	TRC=2...	Transfer request transmitted to the monitor. The monitor rejected the request.
Allocation error	**TRC=3**... + SRC.	Transfer request was transmitted to the monitor. Connect:Express accepted the transfer but an error occured during file processing.
I/O error	TRC=3...	Transfer request was transmitted to the monitor. Connect:Express accepted the transfer but an error occured during file processing.

The following table lists the most frequent return codes. The PRC code is the FTP reply message number.

TRC	PRC	Explanation	Comments
15pp	501 504	Invalid ‘pp’ Parameter	pp: 01 : USER 02 : PASSWORD 09 : PORT 11 : TYPE 12 : STRUCTURE 13 : Transfer MODE 14 : RETRIEVE 15 : STORE 16 : STORE UNIQUE 17 : APPEND 20 : HELP
16pp	500 530	Invalid or unexpected. ‘pp’ Command	pp = 00: unsupported command
2018	553	Store Unique rejected	The file is pre allocated.
2036	553	Security (RACF...)	

TRC	PRC	Explanation	Comments
2040	553	Allocation parameters missing in the Files directory of Connect:Express.	Reception of a file by Connect:Express, allocation rule 2.
2142	553	Partner unknown	
2143	553	File unknown	
2150	553	Invalid protocol	Partner is not a FTP Partner.
2161	553	Invalid IP address	
2208	553	Invalid password	
3001	451	Allocation error	SRC: LOCA: file not found ..
3092	450	LRECL not supported	Active FTP transfer parameters and allocation parameters of the file are not consistent. Example: if STREAM mode, RECORD structure and BINARY data are set, LRECL must correspond to the record built by the FTP client online. Refer to the Connect:Express platform specifications for more details (Data storage and transfer options).

Example error code messages are shown below.

```
  530   FTP1   Logon rejected                                (TRC = 2208)
  553   Transfer rejected                                    (TRC = 2036)
  451   DSN allocation error  SRC=57OC,42CD                 (TRC = 3001)
```

Appendix B

Using Environment Variables

This appendix describes how you can define long names for local and remote files .

Why to Use Long Names

Supporting HFS files means that file names can be much more than 44 characters long. This problem applies to *IP HOST* names, limited to 32 characters long as a Connect:Express parameter fields, and to the *Remote dsn / Pi99* , limited to 44 characters long as a Connect:Express parameter fields.

Connect:Express uses environment variables to support long names. The environment variables are specfic to the monitor, and they are resolved when the name is processed. You can define variables in a file allocated to TOM address space using the ENVVAR DD card . All address spaces that need to process environment variables , such as the APM, the ANM, the AFM and the EAS, know this file.

The fields *Dsn local*, *IP HOST* and *Remote dsn/Pi99* can be extended up to 256 characters. They can contain several variables.

ENVVAR Environment Variables File

The environment variables file is defined in TOM JCL, using DD ENVVAR card. This is a sequential file, fixed format, record length less equal 1024 characters.

```
//SYSIN     DD       DISP=SHR,DSN=PROD.CEXPRESS.SYSIN(TOM4)
//SYJNL     DD       DISP=SHR,DSN=PROD.CEXPRESS.TOM4JNL
//SYLOG     DD       SYSOUT=V,HOLD=YES
//ENVVAR    DD       DISP=SHR,DSN=PROD.CEXPRESS.ENVVAR(TOM4)
```

The environment variable process can issue WTO messages prefixed by ‘ZVAR’, as shown in the example below, where the error is that the file record format is variable.

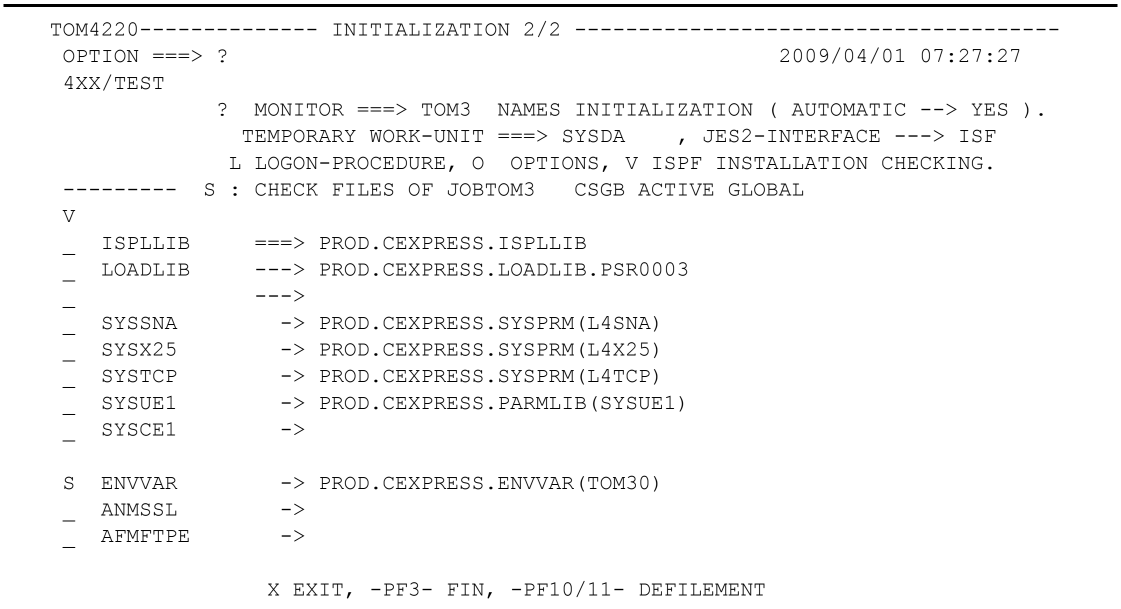
```
ZVAR004E OPEN ENVVAR ERROR  RECFM: 00000080
```

These messages can be issued by any address space that processes variables, APM , ANM, AFM ou EAS. When a syntax error is detected, the process issues a warning and continues. The following example shows that a syntax error has been detected at line 00003 of the ENVVAR file: no ‘>’ character has been found.

```
                 ZVAR006W ENVVAR SYNTAX ERROR   :   NO >    00003
```

If the variable cannot be resolved due to a syntax error, or because no corresponding definition exists, the process continues without resolving it.

Option 0 of TSO/ISPF interface enables the operator to access the ENVVAR file. Use option’S’ to edit the file, updates are dynamically available.



```
TOM4220-------------- INITIALIZATION 2/2 -------------------------------------
 OPTION ===> ?                                          2009/04/01 07:27:27
 4XX/TEST
            ?  MONITOR ===> TOM3  NAMES INITIALIZATION ( AUTOMATIC --> YES ).
              TEMPORARY WORK-UNIT ===> SYSDA    , JES2-INTERFACE ---> ISF
             L LOGON-PROCEDURE, O  OPTIONS, V ISPF INSTALLATION CHECKING.
 ---------  S : CHECK FILES OF JOBTOM3   CSGB ACTIVE GLOBAL
 V
 _  ISPLLIB     ===> PROD.CEXPRESS.ISPLLIB
 _  LOADLIB     ---> PROD.CEXPRESS.LOADLIB.PSR0003
 _              --->
 _  SYSSNA        -> PROD.CEXPRESS.SYSPRM(L4SNA)
 _  SYSX25        -> PROD.CEXPRESS.SYSPRM(L4X25)
 _  SYSTCP        -> PROD.CEXPRESS.SYSPRM(L4TCP)
 _  SYSUE1        -> PROD.CEXPRESS.PARMLIB(SYSUE1)
 _  SYSCE1        ->

 S  ENVVAR        -> PROD.CEXPRESS.ENVVAR(TOM30)
 _  ANMSSL        ->
 _  AFMFTPE       ->

                X EXIT, -PF3- FIN, -PF10/11- DEFILEMENT
```

Syntax and Processing of the Variables

To define a variable in ENVVAR file, use the following syntax :

```
                 <var>value to replace
```

The string ‘var’ can contain 1 to 8 characters , including blanks, excluding ‘>’. The string <var> must be placed in the name you want to extend : ‘value to replace’ will replace it when processed.

Variables are processed in the following situations :

1. •Transfer of an HFS file – data set name processed in APM for PeSIT and Odette, or EAS for Ftp.
2. •TCP/IP connection with host name – processed in ANM or AFM depending on the protocol.

3. •Sending and receiving PeSIT PI99 – processed in APM.
4. •Showing the variable in the TSO/ISPF screens (File definition, Partner definition, Journal)

Message	Description
ZVAR004E OPEN ENVVAR ERROR	The ENVVAR record format is invalid. The cause is indicated. The file transfer is not interrupted, but variables are not resolved
ZVAR006W ENVVAR SYNTAX ERROR	While processing, a syntax error has been detected – the error type and the line number are indicated. All possible variable resolution are completed.
ZVAR007W OVERFLOW, STRING TRUNCATED	While processing, an overflow has been detected : the result is more than 256 characters long : the result is truncated, the process stops.
ZVAR009W VARIABLE NOT FOUND	The variable shown in the message has no definition in the ENVVAR file. All possible resolution are completed

Examples of Use

You can use environment variables in the Dsn local field of the symbolic file definition for an HFS file, in the IP HOST field of the partner definition for a TCP/IP partner , and in the Remote Dsn/Pi99 field of the file definition or the transfer request . You can request the variable resolution in the screen using a command VAR.

Appendix C

FTP Extended Identification

In an FTP environment, 8 characters user names and passwords, used with Connect:Express, are very restrictive. You can use extended parameters and meet requirements of all servers.

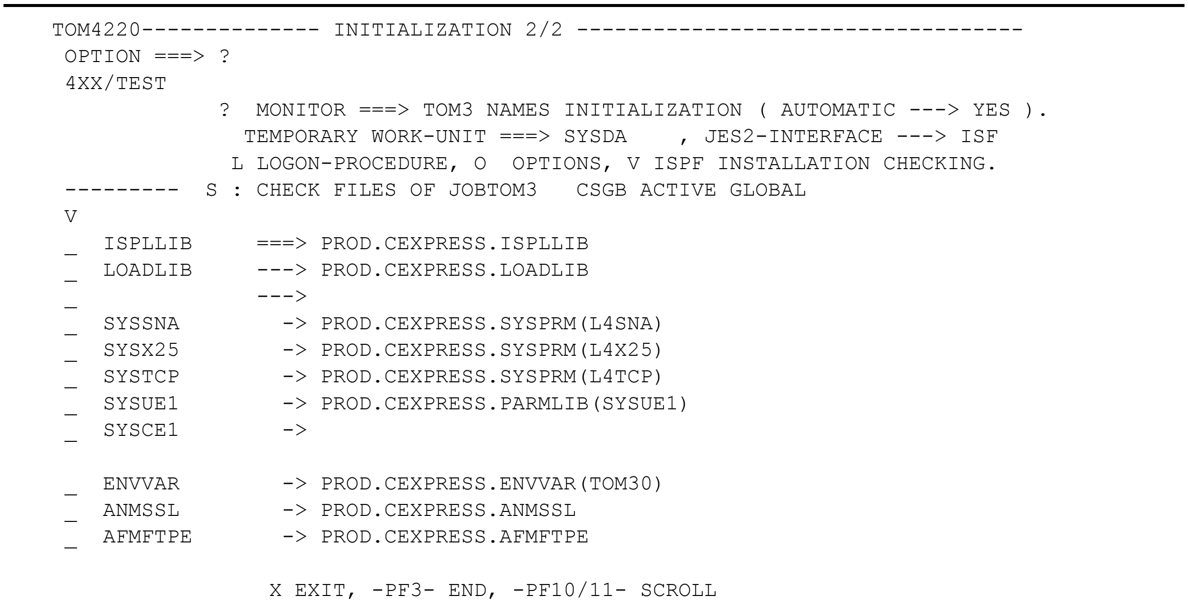
Use the alias field of the FTP partner definition to indicate an identifier for retreiving the extended values. The correspondance is defined in the AFMFTPE file.

AFMFTPE File

The extended identifiers file must be allocated to the AFM address space, using the AFMFTPE DD card. This file is sequential, record format fixed, any record length .

```
//TOMxAFM PROC SSN= (TCP/IP Application, RACF)
//START EXEC PGM=P1AFM000,
// PARM=(T222,&SSN)
//STEPLIB  DD DSN=PROD.CEXPRESS.LOADLIB,DISP=SHR
//         DD DSN=YRESM80.RPTCPIP.V3R1.SEZALINK,DISP=SHR
//SYSABEND DD SYSOUT=V,HOLD=YES
//SYSLOG   DD SYSOUT=V,HOLD=YES
//AFMLOG   DD SYSOUT=V,HOLD=YES,DCB=BLKSIZE=266 LOG file of AFM
//SYSPARM  DD DSN=PROD.CEXPRESS.PARMLIB(AFM&SSN),DISP=SHR
//MIFPARM  DD DSN=PROD.CEXPRESS.PARMLIB(MIF&SSN),DISP=SHR
//AFMFTPE  DD DSN=PROD.CEXPRESS.PARMLIB(FTPE),DISP=SHR
// PEND
```

You can access the AFMFTPE file using option 0 of the TSO/ISPF operator interface.

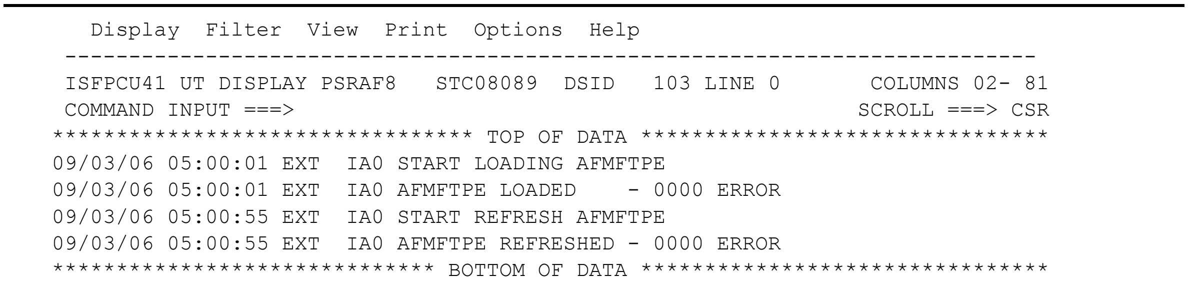


```
TOM4220-------------- INITIALIZATION 2/2 -----------------------------------
 OPTION ===> ?
 4XX/TEST
             ?  MONITOR ===> TOM3 NAMES INITIALIZATION ( AUTOMATIC ---> YES ).
               TEMPORARY WORK-UNIT ===> SYSDA    , JES2-INTERFACE ---> ISF
              L LOGON-PROCEDURE, O  OPTIONS, V ISPF INSTALLATION CHECKING.
 ---------  S : CHECK FILES OF JOBTOM3   CSGB ACTIVE GLOBAL
 V
 _  ISPLLIB     ===> PROD.CEXPRESS.ISPLLIB
 _  LOADLIB     ---> PROD.CEXPRESS.LOADLIB
 _              --->
 _  SYSSNA        -> PROD.CEXPRESS.SYSPRM(L4SNA)
 _  SYSX25        -> PROD.CEXPRESS.SYSPRM(L4X25)
 _  SYSTCP        -> PROD.CEXPRESS.SYSPRM(L4TCP)
 _  SYSUE1        -> PROD.CEXPRESS.PARMLIB(SYSUE1)
 _  SYSCE1        ->

 _  ENVVAR        -> PROD.CEXPRESS.ENVVAR(TOM30)
 _  ANMSSL        -> PROD.CEXPRESS.ANMSSL
 _  AFMFTPE       -> PROD.CEXPRESS.AFMFTPE

                 X EXIT, -PF3- END, -PF10/11- SCROLL
```

The AFM loads the AFMFTPE file during initialization, and if you update the file you have to refresh it, using the following command :

```
/F jobtom,REF FTPE
```

Information messages are issued in the AFM log :



```
   Display  Filter  View  Print  Options  Help
 -----------------------------------------------------------------------------
 ISFPCU41 UT DISPLAY PSRAF8   STC08089  DSID   103 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA *******************************
09/03/06 05:00:01 EXT  IA0 START LOADING AFMFTPE
09/03/06 05:00:01 EXT  IA0 AFMFTPE LOADED    - 0000 ERROR
09/03/06 05:00:55 EXT  IA0 START REFRESH AFMFTPE
09/03/06 05:00:55 EXT  IA0 AFMFTPE REFRESHED - 0000 ERROR
****************************** BOTTOM OF DATA *******************************
```

Syntax of Extended Identification

A line starting by character ‘*’ is a comment line , a blank line is ignored.

A line starting by characters ‘/*’ stops the process (End of file).

A definition must contain a first 8 characters field, followed by a blank

A definition must contain a second field, made of two parameters separated by a comma, or only one parameter.

Below is an example of AFMFTPE file.

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
--
 ISREDDE2   PROD.CEXPRESS.AFM(FTPE)                 - 01.10     Columns 00001 00072
 Command ===>                                                  Scroll ===> CSR
 ****** **************************** Top of Data ******************************
 000001
 000002 *  Test AFMFTPE file
 000003
 000004 DEF002   User_long_25CHAR---------                             Comment
 000005 PARTFTP3 MyUser@address_of_server_destination.com,extended_password
 000006 SPECID   MySpecUser@ address_of_server_destination.com,extended_password_spec
 000007 DEF003   FTP8,FTP4PSW
 000008 DEF005   FTP3Longde25CHAR---------,extendedpassword_17         Comment
 000009 /*
 000010 This line is not processed
 ****** *************************** Bottom of Data ***************************
```

A definition is composed as follows :

```
'IDENTIFIER(8)' extended_user,extended_password
```

The IDENTIFIER is the keyword from which the extended user and password are retrieved. The IDENTIFIER may be the name of a partner in the directory (1 to 8 characters), or any 1 to 6 characters string name that will be referenced by the string $$name

The user name and password are strings without blank inside : the process stops at first blank. If one parameter is provided, this is the user name. The following definition would be rejected

```
000009 DEF006   ,passwordextended
```

How AFMFTPE File is Processed

The AFMFTPE file is loaded during the AFM initialization and each time you request a refresh to Connect:Express.

The syntax is checked and a table is created. In case of error, a message is issued and a record is written in the AFM log. The message indicates the number of the line where the error has been detected. The valid definitions are loaded and a message indicates the number of errors detected.

```
Jesmsglg of AFM
08.00.59 STC09272  FEXT007W AFMFTPE SYNTAX ERROR L=0008

Log of AFM
09/03/06 08:00:58 EXT  IA0 START LOADING AFMFTPE
09/03/06 08:00:58 EXT  IA0 AFMFTPE LOADED    - 0001 ERROR
```

If no definition is valid, the process fails and the following messages are issued :

```
Jesmsglg of AFM
08.12.31 STC09296  FEXT007W AFMFTPE SYNTAX ERROR L=0004
08.12.31 STC09296  FEXT007W AFMFTPE SYNTAX ERROR L=0005
08.12.31 STC09296  FEXT007W AFMFTPE SYNTAX ERROR L=0006
08.12.31 STC09296  FEXT007W AFMFTPE SYNTAX ERROR L=0007
08.12.31 STC09296  FEXT007W AFMFTPE SYNTAX ERROR L=0008
08.12.31 STC09296  FEXT010E AFMFTPE REJECTED 0005/0005 ERRORS
08.12.31 STC09296  AFM-011W AFMFTPE LOADING FAILED - 0005 ERRORS

Log de l'AFM
09/03/06 08:12:30 EXT  IA0 START LOADING AFMFTPE
09/03/06 08:12:30 EXT  AA0 LOAD AFMFTPE FAILED-0005 ERRORS
```

When the refresh fails the old table remains active.

Using Exended Identification

Extended identification is valid for client mode (outbound transfers). Use field DPCSID ALIAS to provide the identifier of the definition 'user,password'.

The Alias parameter is available in the partner definition or in the transfer extension parameters. Use keyword $$FTPE$$ to indicate that the identifier is the partner name itself. Use $$IDENT to indicate that the identifier is IDENT.

If the identiier is not found in the AFMFTPE table, the transfer fails ad the code TRC2219 is returned.

```
Log of the monitor
09/03/06 07:29:06 AFM  OUTGOING FTP SESSION (11/02) REJECTED  WITH FTP8
(010.020.129.003)    RC=2219 TRC=2076 PRC=530

Log of AFM
09/03/06 07:29:05 0001 002 EXTENSION    FTP8     O 010.020.129.003  N:NO MATCH
```

Partner Definition

In the following example, the identifier is PARTFTP3, and the extended values used are (see the AFMFTPE example above) :

MyUser@address_of_server_destination.com,extended_password

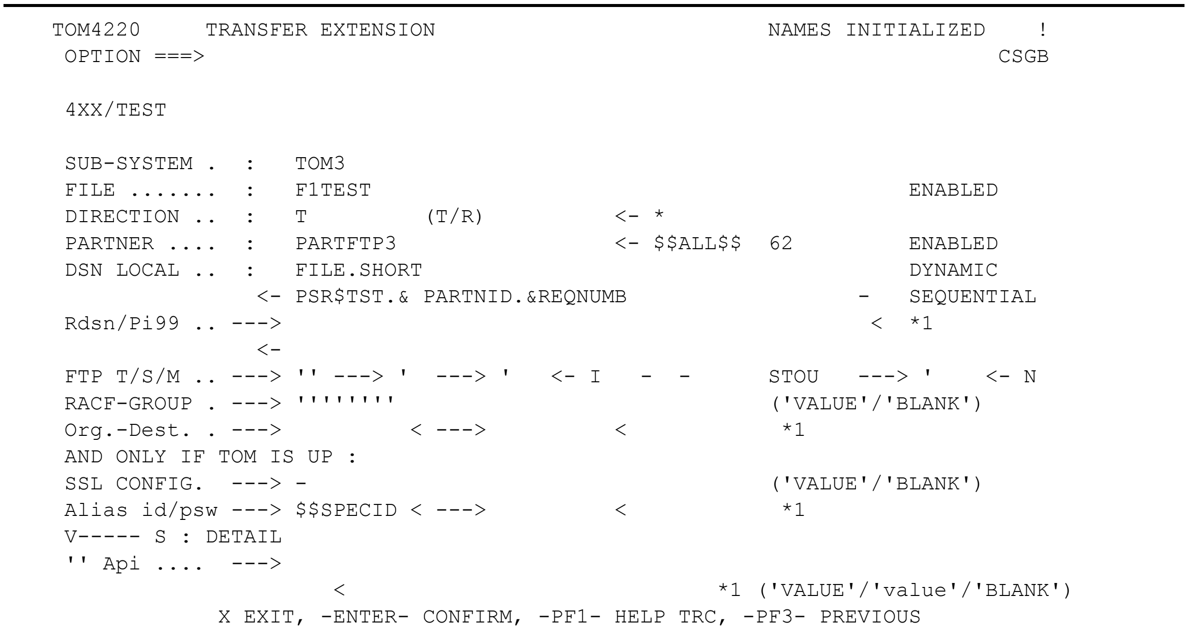
```
TOM4220     PARTNER OF TOM3 TO  UPDATE     (2/4)
 OPTION ===>                        -ENTER- : GO ON, -PF3- : CANCEL   X : EXIT
 TYPE: TOM,FTP
 MOD: PSR0008  09/04/02 04:17:11       68
 SYMBOLIC NAME         : PARTFTP3           DPCSID ALIAS          -> $$FTPE$$
 TOM PASSWORD         => FTP4PSW            DPCPSW ALIAS          -> -
 INITIAL STATE        -> E                  APM RECEPTION CLASS   -> A
 RACF USER            -> TOMPSR             RACF GROUP            -> -

 PARTNER TYPE         => T
 SESSION PROT.NUM.-T. => 6   => 2           SSL CONFIGURATION     -> -
 AUTOMATIC RESTART    -> YES                DN CONTROL MEMBER     -> -

 LINK TYPES           => I   => -
 EFF. TOTAL/IN/OUT    => 128 -> 128 -> 128  FLOW CONTROL T.   SLD -> A

 SNA: LUNAME => -         LOGMODE   -> -        LOGDATA  -> -         DISC -> N
 X25: MCHMSC -> -     REM.ADDR. => -                LOC.ADDR -> -
      CUG    -> -     UDF       -> -                CHARGE   -> -
      FACILITIES      -> -
 IP : ADDR.  => -                PORT => 8108  FTP PASV -> Y PROF.  -> -
      HOST   -> MVSB.<HOST>                           'S': - RIGHTS -> -
 NOTE ->
```

Transfer Request

In the following example, the identifier SPECID is provided in the parameters of the utility

```
P1B2PRQ2.
 000451 SEND SFN=F1TEST,                SYMBOLIC FILE NAME
 000452      SPN=PARTFTP3,              SYMBOLIC PARTNER NAME
 000453      TYP=N,                     REQUEST TYPE
 000455      CLS=A,                     REQUEST CLASS
 000456      PRT=1,                     REQUEST PRTY
 000457      SID=$$SPECID,              DPCSID ALIAS
 000458      DSN=FILE.SHORT
 000459 /*
```

Use the transfer extension screen to provide the identifier in the Alias Id field.



```
TOM4220     TRANSFER EXTENSION                          NAMES INITIALIZED    !
 OPTION ===>                                                             CSGB

 4XX/TEST

 SUB-SYSTEM .  :   TOM3
 FILE .......  :   F1TEST                                         ENABLED
 DIRECTION ..  :   T         (T/R)          <- *
 PARTNER ....  :   PARTFTP3                 <- $$ALL$$  62        ENABLED
 DSN LOCAL ..  :   FILE.SHORT                                     DYNAMIC
               <- PSR$TST.& PARTNID.&REQNUMB                  -   SEQUENTIAL
 Rdsn/Pi99 .. --->                                             <  *1
               <-
 FTP T/S/M .. ---> '' ---> '  ---> '   <- I   -  -      STOU   ---> '    <- N
 RACF-GROUP . ---> ''''''''                             ('VALUE'/'BLANK')
 Org.-Dest. . --->         < --->          <             *1
 AND ONLY IF TOM IS UP :
 SSL CONFIG.  ---> -                                    ('VALUE'/'BLANK')
 Alias id/psw ---> $$SPECID < --->          <             *1
 V----- S : DETAIL
 '' Api ....  --->
                     <                             *1 ('VALUE'/'value'/'BLANK')
             X EXIT, -ENTER- CONFIRM, -PF1- HELP TRC, -PF3- PREVIOUS
```

Index

A

AFM 2-1, 2-3, A-1
- Creating the start procedure 2-3
- Logging file A-1

Allocation rules 2-10

ASCII 2-17

Auxiliary FTP Manager. See also AFM.

B

Batch. See also Interface.

Binary 2-17

C

CC 1-1

Client functions 1-14

Commands
- LIST 1-9
- Modify 2-3
- NLIST 1-9
- Supported by CONNECT:Express 1-2

Commands. See also Interface.

CONNECT:Express
- And FTP 1-2
- Logging file A-2
- Management of FTP transfers 1-13
- Monitor 2-1
- Return codes A-3

Control Connection. See also CC.

CTD connections 1-2

D

Data set name
- Local 2-15
- Remote 2-15

Data structures 2-17
- Using the File option 2-18
- Using the Record option 2-19

Data Transfer Process. See also DTP

Data types 2-17

DSN 2-13

DTP 1-1

E

EAS 2-1

EBCDIC 2-17

Effectors 2-2

Error codes A-3

Exit. See also Interface.

F

File
- Identification 1-13
- Naming 2-14
- Sending and receiving 2-13

Files directory 2-9
- Allocation rules 2-10
- Specific FTP parameters 2-9

FTP
- Client functions 1-14
- Client mode 2-14
- Commands 1-2
- Customizing the FTP list 1-10
- Customizing the welcome message 1-6
- Data structures 2-17
- Data types 2-17
- Defining FTP manager resources 2-4
- Examples of sessions 1-4
- Help 1-6
- Implementing FTP transfers 2-6

Managing sessions 2-8
Messages A-1
Options 1-14, 2-11
Parameters in the SYSIN file 2-6
Protocol options 2-17
Replies issued by CONNECT:Express 1-4
Sending and receiving files 2-13
Server mode 2-14
Transfer modes 2-17

FTP list
Customizing 1-10

FTP manager
Installing 2-1

FTP option
Installing 2-3

FTP protocol 1-1

FTPBYPAS 2-8

I

Identification
File 1-13
Partner 1-13

Installation 2-1

Interface
Monitor 2-5

J

Journal 1-13

L

LIST command 1-9

Local data set name 2-15

Logging file
AFM A-1
CONNECT:Express A-2

M

MIFPARM 2-5

Monitor interface
Defining 2-5

N

Network return codes B-2

NLST command 1-9

O

Operator. See also Interface.

P

PARMFTPL 1-10

Partner
Identification 1-13

Partners 2-7
Managing sessions 2-8
Specific FTP parameters 2-7

PASV 2-9

Pathname 1-3, 2-15

Port number
For the AFM 2-3
Parameter 2-4

PRC A-3

Program. See also Interface.

Protocol return codes. See also PRC.

R

Reception. See also Transfer requests.

Remote data set names 1-14, 2-15

Replies 1-4

Requests. See also Transfer requests.

S

Store Unique 1-15, 2-11, 2-15

SYSIN 2-3, 2-6

SYSPARM 2-1

SYSTCP 2-9

System return codes B-1

T

TCP/IP 1-1, 2-4

Transfer requests
Specific FTP parameters 2-12
To CONNECT:Express 1-14

Translation 2-17

Transmission. See also Transfer requests.

TRC   A-3

U

User interface   1-15