



Connect:Express[®] Unix

FTP Guide
Version 1.4.6

Connect:Express UNIX FTP Guide

Version 1.4.6

Première Edition

This documentation was prepared to assist licensed users of the Connect:Express system ("Sterling Commerce Software"). The Sterling Commerce Software, the related documentation and the information and know-how it contains, is proprietary and confidential and constitutes valuable trade secrets of Sterling Commerce, Inc., its affiliated companies or its or their licensors (collectively "Sterling Commerce"), and may not be used for any unauthorized purpose or disclosed to others without the prior written permission of Sterling Commerce. The Sterling Commerce Software and the information and know-how it contains have been provided pursuant to a license agreement which contains prohibitions against and/or restrictions on its copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright legend.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202-3 with respect to commercial software and commercial software documentation including DFAR 252.227-7013(c) (1), 252.227-7015(b) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), all as applicable.

The Sterling Commerce Software and the related documentation are licensed either "AS IS" or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

References in this manual to Sterling Commerce products, programs, or services do not imply that Sterling Commerce intends to make these available in all countries in which Sterling Commerce operates.

Printed in the United States of America. Copyright © 2003, 2010. Sterling Commerce, Inc. All rights reserved.

Connect:Express is a registered trademark of Sterling Commerce. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

CONTENTS

FTP SUPPORT BY CONNECT:EXPRESS.....	5
INTRODUCTION.....	5
THE FTP PROTOCOL.....	6
<i>FTP commands supported by Connect:Express</i>	7
<i>FTP REPLIES issued by Connect:Express</i>	8
<i>Connect:Express reply to the LIST command</i>	8
CONTROL OF FTP TRANSFER BY CONNECT:EXPRESS	9
<i>Server functions : identification of the file to be transferred.</i>	9
<i>Client Functions : FTP transfer request to Connect:Express</i>	10
<i>User interface</i>	10
<i>General parameters : configuration of the FTP options</i>	11
<i>Partners directory : management of connections</i>	11
<i>Files directory : management of transfers</i>	12
APPENDIX.....	14
<i>File allocation</i>	14
<i>FTP REPLIES by Connect:Express FTP server.</i>	16
<i>TRC return codes.</i>	19
<i>PRC return codes.</i>	20
FTP SUPPORT BY CONNECT:EXPRESS UNIX.....	23
TOM_FTP.....	23
<i>FTP option installation</i>	23
CONNECT:EXPRESS PARAMETERS.....	23
<i>Connect:Express SYSIN file</i>	23
<i>Partners directory</i>	24
<i>Files directory</i>	25
<i>FTP Extended Identification</i>	26
The FTP Extended Identification File.....	26
The FTP DPCSID ALIAS Field	26
<i>Transfer Request parameters</i>	27
APPENDIX.....	28
<i>Files allocation in server mode</i>	28
<i>Storing Data</i>	30

FTP support by Connect:Express

INTRODUCTION

Today LAN's replace clusters of terminals and TCP/IP has become a standard; using FTP as a file transfer protocol has become natural

Strong points of FTP are heterogeneity and wide spreading; no TCP/IP product can be delivered without FTP utility. One of its weak points is the lack of applicative interface allowing to include the file transfer process in an automated exploitation.

To remedy to this FTP applications gap, the FTP protocol has been developed and placed under monitoring of Connect:Express. The goal of this development is to allow applications and users to exchange files in an organized and secured manner, with identified partners, including all facilities which Connect:Express users are accustomed to.

Choices made during FTP implementation under Connect:Express let users use existing FTP applications by allowing them to exchange targeted files (defined in the Files directory) with recognized partners (Those of the Partners directory).

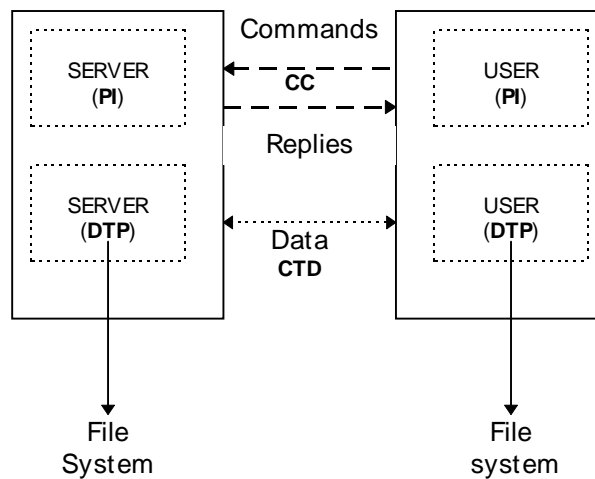
Implementation focuses on a subset of FTP facilities meaningful to accomplish the task of file transfer (logging facilities, Transmission and Reception of a file)

File management (directory management, rename, delete ...) are not supported.

The FTP Protocol

RFC 959 specifies the FTP protocol and refers to RFC 854 (TELNET PROTOCOL SPECIFICATION). It is recommended to read documents of reference for a full comprehension of the subject.

User opens the control connection (CC) with the FTP server (Protocol Interpreter). The exchange of commands (user → server) and replies (REPLY server → user) during this connection allows to identify the user and to negotiate one or several successive file transfers. The data transfer will take place on another connection (CTD) established with the data transfer server (Data Transfer Process).



The diagram above represents the FTP model in the simplest case : negotiation and execution of transfer are done between the same sites.

FTP commands supported by Connect:Express

FTP support by CONNECT:Express is limited to functionalities evoked by commands underlined in the list hereafter. Underlined and italicized FTP commands are not yet supported. FTP commands divide in three types :

a) access control

<u>USER</u>	username
<u>PASS</u> word	password
ACCOUNT	
<u>CWD</u>	(Change Working Directory)
<u>CDUP</u>	(Change to Parent Directory)
SMNT	(Structure MouNT)
<i>REIN</i>	(REINitalize)
<u>QUIT</u>	

b) transfer parameters

<u>PORT</u>	TCP/IP address + port n°
<u>PASV</u>	
<u>TYPE</u>	ASCII (PRINT, <u>NOPRINT</u>) EBCDIC (PRINT, <u>NOPRINT</u>) <u>IMAGE</u> LOCAL byte size
<u>STRU</u>	<u>F</u> ile <u>R</u> ecord Page
<u>MODE</u>	<u>S</u> tream <u>B</u> lock Compressed

c) FTP services

<u>RETR</u> ieve	path
<u>STOR</u> e	path
<u>STOU</u>	path STORe Unique
<i>APPE</i> nd	path
<u>ALLO</u> cate	
<i>REST</i> art	
RNFR	ReName From
RNTO	ReName TO
<i>ABOR</i> t	
DELEte	
RMD	ReMove Directory
MKD	MaKe Directory
<u>PWD</u>	Print Working Directory
<u>LIST</u>	
<u>NLST</u>	Name LiST
SITE	
<u>SYST</u> em	
<u>STAT</u> us	
<u>HELP</u>	item
<u>NOOP</u>	

FTP REPLIES issued by Connect:Express

Replies are built by the server according to usual syntaxes and completed by the own information's of the monitor:

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

Examples of REPLY by the FTP server under Connect:Express are given in annex. On-line help is provided to the FTP client (HELP, SYST, SITE, STAT, PWD commands).

Connect:Express reply to the LIST command

An FTP client has the possibility to ask the server the list of the available files. Connect:Express reply to this demand consists of a list of symbolic names and attributes associated to in the Files directory.

Attributes concern the authorized transfer direction and the allocation of the file on the server:

- A symbolic name can be used for transmission (STOR, STOU, APPE) or for reception (RETR) or both ways.
- A symbolic name can be associated to a partner, to a list of partner or to all partners(\$\$ALL\$).
- Allocation rules defined for a symbolic file can affect demand of transfer (physical name fixed in the directory, pre allocated file...).
- Transfer parameters have to be related to the allocation parameters (Format and length of the record). See, according to the platform, the correspondence table **Data storage and transfer options**.

Control of FTP transfer by CONNECT:Express

FTP Transfers are managed in the same way than any other type of transfer (PeSIT, ETEBAC3, ODETTE...).

A unique request number identifies each FTP transfer running under control of Connect:Express. It is associated to a partner in the Partners directory and a file profile in the Files directory. FTP transfers can be visualized through the operator interface of the monitor (Requests, LOG File, journal). FTP transfers can be automated through the user interface of the monitor (Operator, batch, application).

Server functions : identification of the file to be transferred.

The monitor processes all transfer demands from the three following informations :

- Partner symbolic name
- File symbolic name
- File physical name

The partner symbolic name is fixed during the connection, after controls of the password and the network address.

File physical and symbolic names are obtained during the reception of RETR, APPE, STOR or STOU commands. Each of these two parameters can be fixed by default in the parameters of Connect:Express.

This determines three possibilities for [path], unique parameter of FTP transfer demand:

Identification of the file to be transferred : path =

- *<SYNBNAME!Data-Set-Name>*: the two parameters provided, separated by “ ! ”
- *<Data-Set-Name>*: The symbolic name is taken by default (see C:E parameters)
- *<SYNBNAME!>* : The physical name is the one associated to the symbolic name in the Files directory

SYNBNAME(*optional*) : file name from the monitor Files directory

- Each FTP partner can have a default file profile
- A default profile is defined in the general parameters of Connect:Express.

Data-Set-Name (*optional*): the physical name (**DSN LOCAL**) set in the Files directory is the default.

(Refer to APPENDIX about ‘File allocation’).

Particular case of STOrE Unique (STOU) command

FTP STOU command means that the server does not have to scratch the file if it exists; user expects the server to add a version number to the file name indicated in the STOU command parameter.

Processing of the STOrE Unique command by Connect:Express :

- The allocations rule 1 (pre-allocated file) excludes the use of STOU.
- An index, built from the request number, is added in case of duplicate.

Client Functions : FTP transfer request to Connect:Express

FTP transfer request necessitates, as parameters, a local file name (LOCAL DSN) and a remote file name (REMOTE DSN) or 'path', both implied in the process of transfer.

More, it is necessary to fixe FTP transfer options (transfer mode, data structure, data type).

These parameters (REMOTE DSN, FTP PARAMETERS) may be set in the Files directory of the monitor(They are only used in case of output request). They can also be specified at time of the FTP transfer request.

User interface

The user interface of Connect:Express is the same for all protocols:

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

General parameters : configuration of the FTP options

Start of the FTP monitor : Procedure name
FTP monitor Configuration: Number of sessions, TCP/IP port number...

These parameters are dependent of each platform.

Local identification

Symbolic name and password of the local monitor

Default Profile

It is a symbolic file name, defined in the Files directory of Connect:Express. The default profile specified in general parameters is taken into account for input connections with partners defined with no default profile.

List Option

In server mode, the reception of the FTP 'LIST' command makes Connect:Express send a file containing all the symbolic names under which the online client can undertake transfers. This list is built in function of transmitter partner and receiver partner field of each file in the Files directory.

The option allows to exclude from the list symbolic names to which is associated the keyword \$\$ALL\$\$\$. Only, the symbolic names of file associated to the online partner, or to a list of partners whose it belongs to will be sent in response to the LIST command.

Partners directory : management of connections

Any partner (parameter of the FTP USER command) has to be defined in the partners directory. The session protocol number is 2.

Input connection

During an input connection the FTP USER and PASSWORD parameters are interpreted as symbolic name and partner password, used to identify this one in the Partners directory.

Controls focus on the following parameters :

- Transfer protocol
- Password
- IP address (if any)
- Number of active sessions for the partner
- Status of the partner (enable, disable)

Output connection

The output request for a partner provokes the connection with the distant site. Local identification parameters (symbolic name and password) of the monitor will be used as FTP connection USER and PASSWORD parameters, if no ALIAS name has been defined for the partner.

Default profile

Each partner may be associated a 'default FTP profile' : this symbolic file name has to be defined in the Files directory. The partner default profile is used in case of input connections. If it is not specified in the partner definition, the profile defined in general parameters is taken.

Files directory : management of transfers

A symbolic file name in the Files directory of Connect:Express defines a transfer profile.

This profile allows the management of the transfer by specifying:

- Flows (directions and partners) allowed
- File allocation rules
- File allocation parameters (reception)
- Processing (exits and user commands)
- The remote file name for an output request (transmission and reception)
- FTP transfer options (transfer mode, data structure, data type) for an output request.

The profile name is indicated explicitly during an output request. In case of input request it can be absent : the 'default profile ' defined for the partner (or the default profile of the monitor if none) allows managing such a demand.

Flows:

Usual rules are available (directions, partner transmitter / receiver, list of partners). The FTP **LIST** command received from an FTP client provokes the transfer of the list of symbolic names available for transfer with the online partner.

It is possible to limit this list to the symbolic files explicitly dedicated to the current partner. The full list, without limitation, includes files defined with key words \$\$ALL\$\$ in transmitter and receiver fields.

The list sent to the FTP client indicates allocation rules and transfer direction for each symbolic file.

Allocation rules

Allocation rules works in a standard manner :

- Fixed/dynamic: the physical file name is fixed by the Files directory parameter or can be specified at time of demand.
- Rule 1.2.3 :file has to exist (1), to be created at reception time(2), to be determined by exit user (3).
 ⇒ Rule 1 : TRC allocation error = 3001 if the file does not exist . **On reception the file is overwritten.**
- ⇒ Rule 2: TRC allocation error = 3001 if the file exists, except in case of STOR Unique an index &REQNUMB is added in case of duplicate.
- ⇒ Rule 3: file name passed to the user exit which can modify it and execute the allocation itself.

Allocation parameters

Allocation rule 2 for received files, makes allocation parameters mandatory.

FTP parameters :

- Data type : ASCII (default), EBCDIC or BINARY
- Data structure : FILE (default) or RECORD
- Transfer mode : STREAM (default), BLOCK, COMPRESS

In case of an output request (client mode) the FTP transfer parameters can be fixed in the Files directory or specified at time of the FTP transfer request.

Remote file name:

In case of an output request (client mode) it is possible to specify the 'path' that will be carried by the FTP transfer command. It can be fixed in the Files directory or specified at time of the FTP transfer request.

Appendix

File allocation

In the following :

- '*Symb*' designates a symbolic file name
- '*DSN*' designates a physical file name considered on the host C:E-FTP (local DSN ,remote DSN).
- '*file*' designates the physical name of the file exchanged.

A FTP transfer demand may be formulated in several manners:

STORe (reception by the server), RETRIEve (transmission by the server), STORe Unique (reception by the server with attribution of a version number in case of duplicate), APPENd (reception by the server with data appended to the existent file).

Each demand is specified by a parameter, 'path', which indicates the file name on the server side.

With FTP Connect:Express this 'path' can be constituted of the couple '*Symbolic name!Physical file name*' or one of the two elements :

Path = '*Symb*'!'file' or '*symb*'! or '*file*'

In FTP client mode, the **path** can be specified in the fields '*REMOTE DSN*' in the monitor file directory or at transfer request time.

parameters of the transfer request.

Only the '*Definition type*' is used during the parameters control of the output transfer request .

This control (*FIXED* or *DYNAMIC*) focuses on the '*LOCAL DSN*'; It does not concern the '*REMOTE DSN*'

In FTP server mode the local physical name of the file to be transferred is determined by the monitor from the parameter 'path' of the FTP demand, and under control of the Files directory parameters.

The local physical name is built from the '*LOCAL DSN*' found in the Files directory (entry : SYMB).

Usual keywords (&PARTNID, &FILENAM, &REQ..., &USRVAR1, &USRVAR2, &EXTDSN...) are available.

Using &EXTDSN, &USRVAR1 and &USRVAR2 open the building of the local physical name to the use of the physical file name from the client.

Remark : The 'file' issued from FTP CLIENT can be any character string; FTP server Connect:Express will change it to make it available on the local platform .

Only (&PARTNID, &FILENAM, &REQ..) are available in the 'file' string.

Particular case of the allocation rule 1:

The allocation rule 1 supposes that in reception the file exists and is scratched.

This is contradictory with the STORe Unique command use that allows the writing of duplicate file.

A STORe Unique command received for a file with allocation rule 1 is rejected with the code **TRC=2018**.

Below is summarized the use of symbolic keywords in the LOCAL data set name definition in case of input FTP transfers.

FILE ALLOCATION by Connect:Express FTP SERVER Depending on the LOCAL DSN definition

Reception (STOR, APPE), Transmission(RETR)

FTP Command received	Pathname symb!file	Available Keywords in the LOCAL DSN definition
STOR RETR APPE	symb!	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ
STOR RETR APPE	symb!file	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ; &USRVARi, &EXTDSN: from 'file'
STOR RETR APPE	file	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ &USRVARi, &EXTDSN: from 'file'

Reception (STOU)

FTP Command received	Pathname symb!file	Rule 1	2	3	Available Keywords in the LOCAL DSN definition
STOU	symb!	Rejected TRC=2018	+ <i>REQNUMB</i> (<i>duplicate</i>)	LOCAL DSN defined for 'symb'	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ
STOU	symb!file	Rejected TRC=2018	+ <i>REQNUMB</i> (<i>duplicate</i>)	LOCAL DSN defined for 'symb'	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ &USRVARi, &EXTDSN: from 'file'
STOU	file	Rejected TRC=2018	+ <i>REQNUMB</i> (<i>duplicate</i>)	LOCAL DSN defined for 'symb'	&PARTNID, &FILENAM, &REQ..., &C, &A, &Y, &M, &J, &QQ &USRVARi, &EXTDSN: from 'file'

FTP REPLIES by Connect:Express FTP server.

The following appears on a UNIX FTP client.

- Connect:Express replies are in bold case.
- Commands and parameters from User are underlined.
- Explanations are italicized.

```
ftp> open mvs 4100
Connected to mvs.paris_labs.ig.stercomm.com.
220-Connect:Express For IBM/MVS FTP monitor TOM222 (06/27/97).
220-A STERLING COMMERCE product.
220 Job: STC08854 97/07/01 16:23:58
      OPEN command is answered:
      Version of Connect:Express software (product date)
      The system identification of the C:E FTP server process and the
      connection date and time.
Name (mvs:usr): ftpl
331 FTP1 password please ?
Password: _____
230-FTP1 User logged on at Connect:Express FTP for IBM/MVS.
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 C:E Files directory available for
      User FTP1 is sent.
      A number of File attributes are showed.
      Default profile FTPV is part of the list.
      Following attributes are sent:
      - Dynamic or Fixe Allocation
      - Allocation rule: 2 = to be created, 1 = pre-allocated
      - Direction Transmission, Reception, * = both
      - File record format (Variable, Fixe, 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 for IBM/MVS FTP monitor TOM222 (07/02/97).
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.
```


Help is available on line.

It explains the FTP facilities supported by Connect:Express the general features of FTP transfers monitoring.

```
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 may be ASCII, EBCDIC, IMAGE
214-STRU may be RECORD, FILE
214-MODE may 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 = QQQnnnnnn: 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 C:E special services description.
ftp> rhelp site
214-Connect:Express on IBM/MVS, special services.
214-C:E monitor identifies one transfer by a unique REQUEST number: QQQnnnnnn
214- QQQ = Julian date modulo 186.
214- nnnnnn = Sequence number.
214-
214-Some REPLY shows a TRC return code. This TRC is from C:E 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 and APPE
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:E monitor.
214- A Default profile may be specified by C:E at connection (REPLY 125).
214- If {Profile} is omitted, this Default is kept
214- Dsname is a MVS data set name (1 to 44 characters).
214- C:E monitor associates a MVS Dsname to the Profile in its Directory.
214- If {Dsname} is omitted this Default Dsname is taken.
214- Dsname may contain C:E 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 = MVS 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 will be identified by:
214-REQUEST = QQQnnnnnn 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>
```

Example: transmission to Connect:Express FTP server.

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

      The C:E transfer request number is returned.

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

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

107222 bytes sent in 147,8 seconds (0,7082 kb/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>

      If the transfer is rejected by Connect:Express return codes are issued.
```

TRC return codes.

Current errors are:

- FTP negotiations error: **TRC=15..** (invalid parameter), **TRC=16..** (command rejected), **TRC=11** (invalid command length).

In these cases the demand is not transmitted to the monitor.

- User identification, transfer request rejected by monitor: **TRC=2...**
- Allocation error: **TRC=3... + SRC.**

Monitor has accepted transfer.

- I/O error: **TRC=3...**

Monitor has accepted transfer.

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
2036	553	Security (RACF...)	
2040	553	Allocation parameters missing in the Files directory of Connect:Express.	Reception of a file by C:E, 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 ..
3003	450	Error in Opening Data File	
3092	450	LRECL not supported.	Active FTP transfer parameters and allocation parameters of the file are not consistent. <u>Example:</u> 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).

PRC return codes.

Code	Description
100 Codes	The requested action is being taken. Expect a reply before proceeding with a new command.
110	Restart marker reply.
120	Service ready in (n) minutes.
125	Data connection already open, transfer starting.
150	File status okay, about to open data connection.
200 Codes	The requested action has been successfully completed.
200	Command okay.
202	Command not implemented
211	System status, or system help reply.
212	Directory status.
213	File status.
214	Help message.
215	NAME system type. (NAME is an official system name from the list in the Assigned Numbers document.)
220	Service ready for new user.
221	Service closing control connection. (Logged out if appropriate.)
225	Data connection open, no transfer in progress.
226	Closing data connection. Requested file action successful (file transfer, abort, etc.).
227	Entering Passive Mode
230	User logged in, proceed.
250	Requested file action okay, completed.
257	"PATHNAME" created.
300 Codes	The command has been accepted, but the requested action is being held pending receipt of further information.
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
400 Codes	The command was not accepted and the requested action did not take place. Tthe error condition is temporary, however, and the action may be requested again.
421	Service not available, closing control connection. (May be a reply to any command if the service knows it must shut down.)`
425	Can't open data connection.
426	Connection closed, transfer aborted.
450	Requested file action not taken. File unavailable (e.g., file busy).
451	Requested action aborted, local error in processing.
452	Requested action not taken. Insufficient storage space in system.
500 Codes	The command was not accepted and the requested action did not take place.
500	Syntax error, command unrecognized. This may include errors such as command line too long.

- 501** Syntax error in parameters or arguments.
- 502** Command not implemented.
- 503** Bad sequence of commands.
- 504** Command not implemented for that parameter.
- 530** User not logged in.
- 532** Need account for storing files.
- 550** Requested action not taken. File unavailable (e.g., file not found, no access).
- 552** Requested file action aborted, storage allocation exceeded
- 553** Requested action not taken. Illegal file name.

FTP support by Connect:Express UNIX

Tom_ftp.

The FTP transfers manager is a unique process, started by CONNECT:Express monitor. The program is *strf/tom_ftp* file. It is listening to incoming calls on the TCP/IP PORT number defined in the SYSIN parameters file (**FTP** keyword). The manager is also listening to the monitor and takes into account the interruption and stop commands.

The manager has in charge to accept connections under control of Connect:Express and its Partners directory. Once the connection is established with a partner (FTP control connection), the processing is transmitted at transfer server level.

FTP option installation

Connect:Express for UNIX product is delivered with all its options. To activate an option necessitates getting an authorization key and copy it to the /config directory, with the name *license.key*.

Define the FTPPORT= parameter in the /config/sysin file.

Connect:Express parameters

FTP transfer environment is common with other transfer protocol environments. Some parameters are specific to FTP.

A symbolic file can be shared between FTP transfers and transfers undertaken under the other protocols. Monitoring and all results of the FTP transfers are accessible in the same manner than all the other types of transfer.

Connect:Express SYSIN file

This file contains the Default FTP Symbolic File Name (DEFIL). From 1 to 8 characters, this keyword gives the default file used for FTP transfer in Server mode.

Partners directory

```

C:E/UNIX 146-1 ----- PARTNERS DIRECTORY ----- ce01
OPTION ==>
SYMBOLIC NAME          : FTPPSR
PASSWORD .....        : PSR                PASSWORD OF PARTNER
INITIALIZATION STATUS . : E                E:ENABLE H:DISABLE
PARTNER TYPE .....    : T                T/O
PROTOCOL NUMBER ..... : 2                1:ETEBAC 3, 2:FTP, 3:PESIT
SESSION TABLE NUMBER .. : 6            1->9 SESSION TABLES
X25 PORT .....        : 0                X25 DEVICE NAME
MAX. NO. CONNECTIONS .. : 10/10/10    01->64 TOT/IN/OUT
TYPE OF CONNECTION .... : T                X, P, T OR M
X25 DIAL NUMBER ..... :                1-15 CHARACTERS
LOCAL DIAL NUMBER ..... :                1-15 CHARACTERS
EXTRA NETWORK FIELD ... :
FACILITIES .....     :
TCPIP HOST .....      : localhost          PORT . : 05025
TCPIP ADDRESS .....   :                DEF FTP FILE .. : FTPFILE
DPCSID ALIAS .....   :                SSLPARM ID .... :
DPCPSW ALIAS .....   :                CONTROL OF CERTIFICATES :
NUMBER OF RETRIES .... :                INTERV.SESS ,TRF : , MINUTES
DO YOU WANT TO GO ON ?
OPTION : VIEW          UPD : 08/06/06 10:41 root
-ENTER- NEXT FIELD    -F3- CANCEL          -F8- COMPLETION

```

The connection type is 'T'.

Incoming sessions :

- TCP/ IP address control in case of input call can be inhibited by using SYSTCP (see User's Guide) or by using host name instead of IP address.
- The default file profile is only used in case of input call.

Outgoing sessions:

- The PORT number is only used in case of output call.
- The "21" PORT number is generally the default standard FTP port number.

Files directory

```

C:E/UNIX 146-1 ----- FILES DIRECTORY ----- ce01
OPTION ==>

SYMBOLIC NAME      :      DEFAULT

INITIALIZATION STATUS . : E          E:ENABLE      H:DISABLE
DIRECTION ..... : *          T:TRANSMIT R:RECEIVE *:EITHER
RECEIVING PARTNER ..... : $$ALL$$  'NAME',#LISTE, $$ALL$$
TRANSMITTING PARTNER .. : $$ALL$$  'NAME',#LISTE, $$ALL$$
PRIORITY ..... : 0          0:URGENT 1:FAST 2:NORMAL
DEFINITION TYPE ..... : D          D:DYNAMIC F:FIXED
PRESENTATION TABLE .... : 5          1 -> 9 PRESENTATION TABLE
PARAMETER CARDS FILE  : Y          Y/N
SPACE TO RESERVE ..... : N          Y/N
ALLOCATION RULE ..... : 0          0:INDIF., 1:PREALL., 2:TO CREATE
PHYSICAL NAME ..... : /tmp/TOM_&REQNUMB.tmp
RECORD FORMAT ..... : TV          TF, TV, BF, BU, T*, B*, **
RECORD LENGTH ..... : 00255      1-5 NUMERIC CHARAC.
REMOTE DSN (FTP) ..... : /usr/products/update.txt
TYPE/STRUCTURE/MODE FTP : ARS      E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) .... : N          Y/N FA : N Y/N NOT : (0-7)

OPTION : VIEW          UPD : 08/10/12 14:59 mve
-ENTER- NEXT FIELD    -F3- CANCEL          -F8- COMPLETION

```

Physical name defined in Files directory is taken by default. If definition type is 'Fixed', this name can't be changed at the transfer request; physical name given

PARAMETER CARDS FILE : If set (Y), FTP parameters for transfer can be taken from config/FICPARAMS.dat file.

REMOTE DSN : if set in client mode, this is physical name in remote host.

TYPE/STRUCTURE/MODE (*, File, Stream is supported by all clients) :

- Data Type : ASCII, EBCDIC, Image or Default (*). Default is useful in client mode when FTP remote server software sometimes does not support TYPE command.
- File Structure : File, Record or Default (*). Default is useful in client mode when FTP remote server software sometimes does not support STRU command.
- Transmission Mode : Block, Stream or Default (*). Default is useful in client mode when FTP remote server software sometimes does not support MODE command.

STORE UNIQUE : if set, when client transmitter mode is used, STORE UNIQUE (STOU command) is sent to FTP server.

FTP Extended Identification

FTP Extended identification enables the use of long FTP user names and passwords, in place of the symbolic 8 Characters values, and enables communications through firewalls.

For example, connecting to an FTP server through a firewall is done with this syntax:

```
USER my_username_any_length@server_host_name_any_length
PASSWORD my_password_any_length
```

The FTP Extended Identification File

You can define extended FTP identifications in the apmftpe file placed in the config directory:

```
$TOM_DIR/config/apmftpe
```

You can create and update this file through an editor, using the syntax shown below: Keyword ID is required, commas are used as separators. Extended id or extended password can be omitted.

```

ID=Aliasid,Ftp_Extended_Id,Ftp_Extended_Password
or
ID=Aliasid,Ftp_Extended_Id (no password)
or
ID=Aliasid,,Ftp_Extended_Password (no user name)
```

The FTP DPCSID ALIAS Field

The partner definition provides the DPCSID ALIAS field. If this field is set with \$\$FTPE\$\$ keyword, the alias id must be picked up in the apmftpe file, identified by the partner name. If the DPCSID ALIAS field is set with a name starting with '\$\$', \$\$alias for example, the alias id must be picked up in the apmftpe file, identified by the name "alias" from \$\$alias.

```

C:E/UNIX 146-1 ----- PARTNERS DIRECTORY ----- tom1
OPTION ==>

SYMBOLIC NAME ..... : PFTP01
PASSWORD ..... : PASSWD          PASSWORD OF PARTNER
INITIALIZATION STATUS . : E          E:ENABLE      H:DISABLE
PARTNER TYPE ..... : O          T/O
PROTOCOL NUMBER ..... : 2          1:ETEBAC 3, 2:FTP, 3:PESIT
SESSION TABLE NUMBER .. : 1          1->9 SESSION TABLES
X25 PORT ..... :
MAX. NO. CONNECTIONS .. : 20/10/10  01->64 TOT/IN/OUT
TYPE OF CONNECTION .... : T          X, P, T OR M
X25 DIAL NUMBER ..... :
LOCAL DIAL NUMBER ..... :
EXTRA NETWORK FIELD ... : 'USER-DATA-FIELD'
FACILITIES ..... :
TCPIP HOST ..... : localhost          PORT . : 05050
TCPIP ADDRESS ..... :
DPCSID ALIAS ..... : $$FTPE$$      SSLPARM ID .... :
DPCPSW ALIAS ..... : PASSWD        CONTROL OF CERTIFICATES :
NUMBER OF RETRIES ..... :
DO YOU WANT TO GO ON ?
OPTION: VIEW          UPD: 10/04/01 15:33 gcz
-ENTER- NEXT FIELD   -F3- CANCEL      -F8- COMPLETION
```

For PFTP01, the apmftpe file will provide the ID=PFTP01 information.

```
ID=PFTP01,Ftp_Extended_pftp01,Ftp_pftp01_Extended_Password
ID=alias,my_alias,my_alias_Password
ID=SPECID,my_user@address_of_server_destination.com
```

You can pass the DPCSID ALIAS with the transfer request, from stern or itom. For example:

```
$TOM_DIR/itom/p1b8preq "/SFN=FILE01/DIR=T/SPN=PFTP01/SID=$$SPECID"
"/DSN=\$TOM_DIR/config/sysin"
```

For this request, the apmftpe file will provide the ID=SPECID information.

If a field is omitted in the apmftpe file for the current definition, the value is taken from the directory, or the default sysin value. For example SPECID does not have a password: the password is PASSWD from the directory DPCPSW ALIAS field. If no extended user id is provide, the DPCSID name from the sysin is used: this is the way to just providing an extended password.

Transfer Request parameters

```
C:E/UNIX 146-1 ----- TRANSFER REQUEST ----- ce01
OPTION ==>
FILE ..... : SYMBFILE          DIRECTION ..... : T (T/R)
PARTNER ..... : PARTNER.
DPCSID ALIAS ..... : MYNAME..      DPCPSW ALIAS ..... : MYPASSW.
ORIGIN..... :                      DESTINATION..... :
SENDER..... :                      RECEIVER ..... :
PHYSICAL NAME ..... : /usr/lib/libsock.a.....
REMOTE PHYSICAL NAME .. : PHYSICAL.FILENAME.TRANSMIT.....
LABEL:.....
RECORD FORMAT ..... : BU              TF, TV, BF, BU
RECORD LENGTH ..... : 02048
TYPE/STRUCTURE/MODE FTP : ***          E/A/I/* ,F/R/* ,B/S/*
STORE UNIQUE (FTP) .... : N            Y/N FA : N Y/N NOT : (0-7)
TYPE ..... : N            (N/I/H)
TYPE OF CONNECTION .... : T            (X/P/T)
PRIORITY ..... : 0            (0/1/2)
DATE ..... : 19980728101604 (YYYYMMDDHHMMSS)
API FIELD (ETEBAC3 : 80 CHARACTERS FOR CARD)
1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0
DO YOU WANT TO GO ON ?
-ENTER- NEXT FIELD          -F3- CANCEL          -F8- COMPLETION
```

Appendix

Files allocation in server mode

In the following:

- '**Symb**' designates a symbolic file name
- '**DSN**' designates a physical file name considered on the host C:E-FTP (local DSN ,remote DSN).
- '**file**' designates the physical name of the file exchanged.

An FTP transfer demand may be formulated in several manners:

STORe (reception by the server), RETRIEve (transmission by the server), STORe Unique (reception by the server with attribution of a version number in case of duplicate), APPEnd (reception by the server with data appended to the existent file).

Each demand is specified by a parameter, 'path', that indicates the file name on the server site.

With FTP Connect:Express this 'path' can be constituted of the couple '*Symbolic name!Physical file name*' or one of the two elements :

Path = 'Symb'!'file' or 'symb'! or 'file'

In FTP server mode the local physical name of the file to be transferred is determined by the monitor from the parameter 'path' of the FTP demand, and under control of the Files directory parameters.

The local physical name is built from the 'LOCAL DSN' found in the Files directory (entry : SYMB). Usual keywords (&PARTNID, &FILENAM, &REQNUMB, &USRVAR1, &USRVAR2, &EXTDSN... are available).

Using &EXTDSN, &USRVAR1 and &USRVAR2 open the building of the local physical name to the use of the physical file name from the client.

Remark : The 'file' issued from FTP CLIENT can be any character string; FTP server Connect:Express will change it to make it available on MVS system .

Only (&PARTNID, &FILENAM, &REQNUMB) are available in the 'file' string.

Particular case of allocation rule 1:

Allocation rule 1 supposes that in reception the file exists and is scratched.

This is contradictory with the STORe Unique command use that allows the writing of duplicate file.

A STORe Unique command received for a file with allocation rule 1 is rejected with the code **TRC=2018**.

In server mode the local file physical name to be transferred is determined by the monitor from the 'path' parameter of received demand, and under control of the parameters found in the Files directory.

Example:

Reception of FTP command from some remote system:

STOR FTEST!ApplicationName/File.dat

- 'symb' and 'file' are given.
- The symbolic file is: **FTEST**

Processing by CONNECT: Express FTP server for Unix:

FTEST:

DSN LOCAL = /home/tmp/&EXTDSN

- The variable &EXTDSN is replaced by physical name sent by Ftp client application.

Resulting data set name is =

/home/tmp/ApplicationName/File.dat

Notes that ApplicationName is a directory and must exist.

FTEST:

DSN LOCAL = /home/tmp/ &PARTNID.&REQNUMB

The 'file' string received is ignored (no &EXTDSN, &USRVARi keyword)

Resulting data set name is =

/home/tmp/PARTFTP.A1700004

Storing Data

Depending on:

- FTP transfer parameters, **transfer MODE - Data TYPE - Data STRUCTURE**
- Record format of the file allocated on Unix (FIXED or VARIABLE format) data received via FTP by AFM will be processed in different ways:
- Data are submitted automatically to translation or not.
- The file structure allocated for reception (RECFM, LRECL) has to allow the storing of data and respect their structure when it is specified.

Data type Influence.

Three types of data are supported:

- *ASCII*: text files structured by separators, not submitted to translation; separator X'0D0A' are suppressed, the LRECL of the file has to be sufficient.
- *EBCDIC* : text files structured by separators, submitted to translation; separator X'15' are suppressed, the LRECL of the file has to be sufficient.
- *BINARY*: Data are not submitted to any processing before storing; (see hereafter data structure influence)

Data Structure Influence.

Two data structures are supported:

- *File* (FILE): record separator, if any, is part of data; for binary data, with no separator, they are segmented and stored in the file.
- *Record* (RECORD): record separator is part of protocol information; binary type data are stored by respecting the transportation record size (determined by the transmitter software): the file has therefore to be allocated consequently.

The processing is different for binary type data only, since ASCII or EBCDIC text files are implicitly structured.

Transfer Modes .

The transfer mode (STREAM, BLOCK) do not influence the way data are processed .

