English

FUJITSU

openFT V12.0 for z/OS

Managed File Transfer in the Open World

User Guide

Edition September 2012

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

1 Introduction . . . 11

1.1 Brief description of the product openFT for z/OS . . . 12
1.2 Target group . . . 12
1.3 Concept of the openFT for z/OS manuals . . . 13
1.4 Changes since the last version of the manual . . . 14
1.5 Notational conventions . . . 15
1.6 License provisions . . . 16

2 openFT - the Managed File Transfer . . . 19

2.1 Heterogeneous computer systems . . . 21
2.1.1 File conversion . . . 21
2.1.2 openFT product range . . . 22
2.2 Heterogeneous networks . . . 24
2.2.1 openFT partners . . . 24
2.2.2 FTP partners . . . 25
2.3 Transferring files . . . 27
2.3.1 Specifying the transfer start time . . . 28
2.3.2 Controlling the duration of a request . . . 28
2.3.3 Request queue . . . 29
2.3.4 Automatic restart . . . 30
2.4 File management . . . 31
2.5 Remote command execution . . . 32
2.6 Automation . . . 33
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing . . . 33
2.6.1.1 Preprocessing . . . 34
2.6.1.2 Postprocessing . . . 34

2.6.1.3 Follow-up processing . . . 34
2.6.2 Program interfaces . . . 36
2.7 Further processing of openFT data . . . 37
2.8 Secure operation . . . 38
2.8.1 The FTAC function . . . 38
2.8.1.1 Features of the FTAC function . . . 38
2.8.1.2 Admission set . . . 40
2.8.1.3 FT profile (admission profile) . . . 40
2.8.1.4 Effects of an admission profile . . . 43
2.8.1.5 FTAC administrator . . . 44
2.8.2 Encryption for file transfer requests . . . 44
2.8.3 Logging openFT operations - the logging function . . . 46
2.8.4 Authentication . . . 48
2.9 Using openFT in a Sysplex composite . . . 50

3 File transfer and file management . . . 51

3.1 File names . . . 52
3.1.1 Unique file names for receive files . . . 52
3.1.2 BS2000/OSD file names . . . 54
3.1.3 File names in Unix systems . . . 56
3.1.4 Windows file names . . . 56
3.1.5 z/OS file names . . . 57
3.2 File passwords . . . 61
3.3 File types . . . 62
3.3.1 BS2000/OSD files . . . 62
3.3.2 z/OS files . . . 63
3.3.2.1 Files that can be transferred . . . 65
3.3.2.2 Volumes . . . 65
3.3.2.3 Attributes of receive files . . . 66
3.3.2.4 Transferring a PO or PDSE member . . . 79
3.3.2.5 Transferring a PO or PDSE data set . . . 81
3.3.2.6 Transferring a generation data set . . . 85
3.3.3 Unix and Windows files . . . 89
3.3.4 Transfer of various file types . . . 93
3.3.5 Migrated files . . . 95

3.4 Transferring 7-bit, 8-bit and Unicode files . . . 96
3.4.1 Code tables and coded character sets (CCS) . . . 96
3.4.2 Specifying the CCS on a transfer request . . . 97
3.4.3 Data conversion . . . 98

3.5 Entries for the remote system . . . 99
3.5.1 Defining the partner computer . . . 99
3.5.2 Transfer admission . . . 102

3.6 Options for file transfer . . . 104
3.6.1 Maximum record lengths . . . 104
3.6.2 Syntax rules . . . 104
3.6.3 Compressed file transfer . . . 107
3.6.4 Encrypted file transfer . . . 107
3.6.5 Notifying results . . . 108
3.6.5.1 Messages and return codes automatically issued by openFT for z/OS . . . 108
3.6.5.2 Result lists generated by openFT for z/OS . . . 110
3.6.5.3 User-generated result information . . . 110
3.6.6 Preprocessing and postprocessing . . . 111
3.6.7 Follow-up processing . . . 114
3.6.8 Accounting of file transfer requests . . . 125

3.7 File management . . . 127
3.7.1 File management of z/OS files with openFT for Windows . . . 128
3.7.2 Displaying file attributes . . . 132
3.7.3 Displaying directories . . . 135
3.7.4 Renaming files . . . 136
3.7.5 Renaming directories . . . 137
3.7.6 Deleting files . . . 137
3.7.7 Deleting directories . . . 138

4 Menu interface for the FT user . . . 139

4.1 Creating an openFT instance . . . 139

4.2 General . . . 140

4.3 Software requirements . . . 141

4.4 Representation and utilization . . . 142

4.5 Error messages . . . 146

5 Command interface 147

5.1 Setting an openFT instance 148

5.2 Functional command overview 149
5.2.1 FT command overview 149
5.2.2 FTAC commands overview 151

5.3 Entering FT commands 152

5.4 Command syntax representation 155

5.5 Command return codes 163

5.6 Output in CSV format 164

5.7 FTCREDIR
Create remote directory 165

5.8 FTCREPRF
Create admission profile 168

5.9 FTDEL
Delete remote files 188

5.10 FTDELDIR
Delete remote directory 191

5.11 FTDELPRF
Delete admission profile 194

5.12 FTEXEC
Execute remote command 197

5.13 FTHELP
Display information on reason codes in the logging records 202

5.14 FTMOD
Modify remote file attributes 204

5.15 FTMODADS
Modify admission set 209

5.16 FTMODDIR
Modify remote directory attributes 215

5.17 FTMODPRF
Modify admission profile 218

5.18 FTMODREQ
Modify request queue 240

5.19 FTSCOPY
Transfer file synchronously . . . 243

5.20 FTSHW
Display remote file attributes . . . 248

5.21 FTSHWADS
Display admission sets . . . 254

5.22 FTSHWINS
Display an openFT instance . . . 258

5.23 FTSHWLOG
Display log records and offline log files . . . 259
5.23.1 Description of the short output . . . 269
5.23.2 Description of the long output . . . 271

5.24 FTSHWMON
Show monitoring data . . . 276
5.24.1 Description of the monitoring values . . . 279
5.24.2 Examples . . . 284

5.25 FTSHWOPT
Display operating parameters . . . 287
5.25.1 Description of the output . . . 289

5.26 FTSHWPRF
Display admission profile . . . 294

5.27 FTSHWPTN
Display partner systems . . . 298

5.28 FTSHWRGE
Display partner systems . . . 305

5.29 NCANCEL
Cancel file transfer requests . . . 308

5.30 NCOPY
Transfer file asynchronously . . . 312
5.30.1 Introduction to the NCOPY command . . . 312
5.30.1.1 The shortest form of the command . . . 313
5.30.1.2 How to find out if the file transfer request has been executed . . . 313
5.30.2 Full form of the NCOPY command . . . 316
Specifications for the local system (LOCAL-PARAMETER) . . . 320
Follow-up processing in the local system . . . 324
Specifications for the remote system (REMOTE-PARAMETER) . . . 327
Follow-up processing in the remote system . . . 336
Optional entries . . . 338

5.30.3 Examples of the NCOPY command . . . 345

5.31 NSTATUS
Query status of file transfer request . . . 353
5.31.1 Description of the short output . . . 359
5.31.2 Description of the long output . . . 360
5.31.3 Description of the summary output . . . 364

6 Program interface for the FT user . . . 365

6.1 Macro OPENFT to call a user command . . . 367

7 What to do if 371

7.1 Frequently asked questions . . . 376

7.2 Reporting errors . . . 379

8 Appendix . . . 381

8.1 Structure of CSV outputs . . . 381
8.1.1 Output format . . . 381
8.1.2 FTSHW . . . 383
8.1.3 FTSHWADS . . . 385
8.1.4 FTSHWLOG . . . 387
8.1.5 FTSHWMON . . . 390
8.1.6 FTSHWOPT . . . 394
8.1.7 FTSHWPRF . . . 399
8.1.8 FTSHWPTN . . . 403
8.1.9 FTSHWRGE . . . 405
8.1.10 NSTATUS . . . 406

8.2 FT system messages . . . 411
8.2.1 FTR messages . . . 413
8.2.2 FTC messages . . . 453

8.3 Using openFT in z/OS systems without the TSO interactive system . . . 464

Glossary 465

Abbreviations 487

Additional documentation 491

Index 493

1 Introduction

The openFT product range transfers and manages files

– automatically,
– securely, and
– cost-effectively.

The reliable and user-friendly transfer of files is an important function in a high-performance computer network. The corporate topologies consist of networked PC workstations, which are usually additionally linked to a mainframe or Unix based server or Windows server. This allows much of the processing power to be provided directly at the workstation, while file transfer moves the data to the mainframe for further processing there as required. In such landscapes, the locations of the individual systems may be quite far apart. Fujitsu Technology Solutions offers an extensive range of file transfer products - the openFT product range - for the following system platforms:

- BS2000/OSD®
- Solaris™ (SPARC®/Intel™), LINUX®, AIX®, HP-UX®
- Microsoft® Windows Vista™, Windows™ 7, Windows Server 2008™ and Windows Server 2008 R2™
- z/OS (IBM®)

1.1 Brief description of the product openFT for z/OS

openFT for z/OS is the file transfer product for computers using the operating system z/OS.

All openFT products communicate with each other using the openFT protocol (previously known as FTNEA) as laid down by Fujitsu. Since a number of FT products from other software suppliers also support these protocols, many interconnection options are available.

openFT supports the TCP/IP and SNA transport protocols.

The range of functions made available by openFT can be extended using the add-on products openFT-FTP and openFT-AC:

- openFT-FTP supports FTP functionality.
- openFT-AC provides extended system and data access protection. FTAC stands for File Transfer Access Control.

1.2 Target group

This manual is intended for z/OS users who want to run the openFT (File Transfer) product.

It describes how to transfer files between two systems and how to make file transfer safer using openFT-AC.

To understand this manual, it is necessary to have a knowledge of the z/OS operating system.

1.3 Concept of the openFT for z/OS manuals

The openFT for z/OS product with its optional components openFT-FTP, openFT-AC and openFT-CR is described in two manuals. In addition to this User Guide, there is also a System Administrator Guide "openFT for z/OS - Installation and Administration”.

The manuals are arranged as follows:

- openFT for z/OS - Managed File Transfer in the Open World

  The user guide contains the following information:

  – an overview of the basic functions of the openFT product family
  – a detailed description of the conventions for file transfer to computers with different operating systems
  – a description of the user commands and the menu and program interface for the FT user
  – the openFT and openFT-AC messages for the FT user

- openFT for z/OS - Installation and Administration

  The System Administrator Guide is aimed at the FT administrator and the FTAC administrator. It describes the following:

  – how to install openFT and its optional components, including the requirements for using the product
  – how to operate, control and monitor the FT system and the FTAC environment
  – the administration commands for the FT administrator, the FTAC administrator and the remote administrators and also the menu and program interface
  – the openFT and openFT-AC messages for the FT administrator
  – additional sources of information for the FT administrator, such as the account records and the logging information

You will also find current information on the Internet under *http://de.ts.fujitsu.com/openft* (german) or *http://ts.fujitsu.com/openft* (english).

1.4 Changes since the last version of the manual

Compared to the User Guide for openFT V11.0 for z/OS, the User Guide for openFT V12.0 for z/OS describes the following new features:

Extended logging functions

The logging functions have been extended as follows:

- Switch log file and offline logging

  The FT administrator can switch the log file during operation. After switchover, new log records are written to a new log file. The previous log file is retained as an offline log file. The log records it contains can still be viewed using the tools available in openFT. To permit this, the command FTSHWLOG has been extended:

  – New operands LOGGING-FILE and PREVIOUS-FILES that make it possible to view log records from offline log records.

  – New operand value INFORMATION=*LOGGING-FILES to output the names of all log files (including offline log files).

- Polling function for the output of log records

  In FTSHWLOG, the new operand NUMBER=*POLLING can be used to set the interval and number of repetitions (polling).

- Wildcards for partner names during the output of log records

  In FTSHWLOG, it is also possible to use the wildcards "*" and "?" when specifying the partner name.

Enhanced security functions

Authentication level 2 for the public keys of partner systems meets higher security requirements. FTSHWLOG displays the authentication level (output parameter SEC-OPTS, new values LAUTH2 and RAUTH2).

Extended partner management

- The FT administrator can now also explicitly deactivate partners in the partner list for inbound requests. In FTSHWPTN, this attribute is displayed in the output parameter INBND .
- The FT administrator can control whether asynchronous outbound requests to a given partner should always be run serially or whether parallel connections are also permitted. In the FTSHWPTN command, this attribute is displayed in the output parameter REQU-P.

Extended request management

- Global request ID

  In the event of an FT request, the initiator's request number is transferred to the responder where it is visible as a global request ID. This means that any request can be unambiguously assigned to an initiator and responder.

  The NSTATUS and FTSHWLOG commands have been extended as follows:

  – At the responder, the global request ID is displayed in the new output parameter GLOB-ID in each command.

  – The new parameter GLOBAL-REQUEST-ID makes it possible to perform selection on the basis of a global request ID in both commands.

Other changes

- The maximum value for the TRANSFER-ID (request number) that can be specified in a number of different commands has been changed to 2147483647.
- The description of dynamic partners is now more precise. To this end, the partner types "named partner", "registered dynamic partner" and 'free dynamic partner" have been introduced.
- The description of the CSV output for the SHOW commands (FTSHWxxx and NSTATUS) has been greatly extended.

1.5 Notational conventions

The following notational conventions are used throughout this manual:


indicates notes


Indicates warnings.

Additional conventions are used for the command descriptions, see section “Command syntax representation” on page 155.

1.6 License provisions

The following provisions apply to the use of Secure FTP.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

```
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit. See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact opens-
sl-core@openssl.org.

OpenSSL License
---------------

/* ====================================================================
Copyright (c) 1998-2006 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must
   display the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in
   the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse
   or promote products derived from this software without prior written permis-
   sion. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may
   "OpenSSL" appear in their names without prior written permission of the
   OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledg-
   ment:
   "This product includes software developed by the OpenSSL Project for use in
   the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
```

SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
==

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

2 openFT - the Managed File Transfer

Managed File Transfer is a term that documents the high performance of openFT products. Such high demands on corporate file transfer result, on the one hand, from the variety of hardware and software commonly installed today and, on the other, from the different needs your company has with respect to file transfer itself. A further important aspect of enterprise file transfer is provided by the options for automation and the security functions offered by openFT. In addition, central administration of an openFT network and presentation of the operating states make openFT a managed file transfer system.

Fujitsu Technology Solutions offers a comprehensive openFT product range for Managed File Transfer, which can be used to operate **heterogeneous computer systems** (hardware and software) of many manufacturers ranging from mainframe systems to the PC. openFT products can be used in various operating systems such as Windows, Unix systems, BS2000/OSD, z/OS and others.

Even **heterogeneous networks** such as TCP/IP, NEA, ISO-FTAM, X.21/X.25, ISDN and GSM mobile telephony or MODACOM pose no problem for openFT. The continual integration of new platforms and network types guarantees high availability of the openFT products, also in the future. Not all networks are supported on all platforms.

The integration of the **ISO 8571 FTAM standard** (File Transfer, Access and Management) guarantees uniform interfaces for requests to openFT partners and any FTAM partners (not available under z/OS).

Support for the **FTP protocol** makes it possible to connect to FTP servers and FTP clients on any required platform.

Functions such as request storage, automatic restart, job and file management, follow-up processing, resource management, program interfaces, encryption and authentication indicate the wide range of services offered by openFT products, thus making them truly suitable for Managed FIle Transfer.

Request storage makes it possible to start **asynchronous file transfer** at any desired time, e.g., to save charges or to wait for the occurrence of specific events. The **automatic restart** feature ensures a consistent continuation of file transfer after the correction of a fault, e.g., a network or processor failure.

Automation is achieved, among other things, via facilities for preprocessing and follow-up processing:

- Local or remote **preprocessing** enables data to be created within a send or receive request by starting a job, for example, and then transferring it then to the local or remote system.
- Local or remote **postprocessing** enables the data transferred to be processed further within a send or receive request.
- Preprocessing as well as postprocessing can be executed within a request.
- **Follow-up processing** permits any job to be started just after file transfer. You can make the start of follow-up processing dependent on the success of the file transfer.

The **program interfaces** permit the implementation of openFT functions in programs.

File management in the remote and local systems provides facilities for modifying file attributes. for example.

The **resource control** allows you to store file transfer requests at any time and have them issued automatically when the partner system is available. The use of Monitor Job Variables in BS2000/OSD is also possible.

In the case of **synchronous file transfer**, you must wait until data transfer has been completed and you can then immediately react to the result.

Protection of the data inventory is becoming a priority issue in companies in view of the open nature of today's networks. The **FTAC functionality** (optional in openFT for BS2000/OSD and openFT for z/OS) integrated in openFT products offers comprehensive and individually scalable protection functions:

– decoupling of transfer admissions and login admission
– access rights dependent on the partner systems
– user-specific access rights
– flexible access right levels
– recording of every authorization check

The **logging** of data transfer requests and authorization checks permits evaluation of previous request and access, thus providing a further security feature.

The **encryption** of request description and transfer data is another protection level provided by openFT. Request description data include the authorization data for the transfer of and access to data (e.g. transfer admission, file password). In addition, it is possible to connect to system security functions such as SECOS on BS2000, RACF and ACF2 on z/OS.

Expanded identity checking (i.e. **authentication**) of the communications partner is offered for requests involving openFT partners. It is based on addressing network-wide, unique IDs for openFT instances and the exchange of partner-specific key information.

2.1 Heterogeneous computer systems

One strength of the openFT products is their capability for linking different computers, particularly computers from different manufacturers running various operating systems. The precondition for file transfer between two computers is that a transport connection exists between these two computers and that one of the openFT products or an FTP application is installed on the computers.

The openFT products are matched for optimum interoperability. They retain file structures and attributes during file transfer. openFT products cannot override the conventions that apply to the operating system. Data conversion may be necessary to ensure that characters are represented correctly when performing transfers between certain operating systems.

2.1.1 File conversion

The coding, i.e. the system-internal representation of individual characters, letters and digits, depends on the operating system. The data must then be converted because

– Internally, Unix and Windows computers use an ASCII-based code (American Standard Code for Information Interchange). For Unix systems this is an ISO-8859-x code that is descried in ISO standard 8859. For Windows systems, this is a code defined by Microsoft such as, for example, the CP1252 character set with Euro symbol for western Europe.
– BS2000/OSD systems and z/OS computers, on the other hand, normally use an EBCDIC (Extended Binary-Coded Decimal Interchange Code).

Data conversion between openFT partners always applies to the characters with which parameter values (e.g. file names, user IDs, follow-up processing strings, etc.) are transferred.

The conversion of file contents, by contrast, is only relevant for files to be transferred in text format; no data conversion is performed by openFT when transferring files in other formats (binary, transparent, etc.).

Please note that the openFT partner codes use the same character repertoire. If this is not the case, some of the characters in the text file (e.g. umlauts) may not be represented correctly. If you transfer files with openFT partners as of V10, you can assign the "Coded Character Sets" that are to be used for local and remote data conversion in the request. It is also possible to transfer Unicode files with these partner systems, see section “Transferring 7-bit, 8-bit and Unicode files” on page 96.

2.1.2 openFT product range

The tables below provide an overview of the openFT product range, showing the openFT products currently available for your computer.

openFT product range

Product	Operating system	Comment
openFT for Unix systems	AIX, Linux, HP-UX, Oracle Solaris	Additional systems on request
openFT for BS2000/OSD	BS2000/OSD	BS2000 systems from Fujitsu Technology Solutions
openFT for Windows systems	Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7	Intel architecture
openFT for z/OS	z/OS	z/OS systems from IBM

openFT add-on products

Product/delivery unit	Operating system	Comment
openFT-FTAM for Unix systems	AIX, Linux, HP-UX, Oracle Solaris,	Unix systems
openFT-FTAM for Windows systems	Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7	Intel architecture
openFT-FTAM for BS2000/OSD	BS2000/OSD	FTAM functionality forBS2000 systems from Fujitsu Technology Solutions
openFT-FTP for Unix systems	AIX, Linux, HP-UX, Oracle Solaris	Unix systems
openFT-FTP for Windows systems	Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7	Intel architecture
openFT-FTP for BS2000/OSD	BS2000/OSD	FTP functionality for BS2000 systems
openFT-FTP for z/OS	z/OS	FTP functionality for z/OS systems
openFT-AC for BS2000/OSD	BS2000/OSD	FTAC functionality for BS2000 systems
openFT-AC for z/OS	z/OS	FTAC functionality for z/OS systems
openFT-CR	All platforms of the openFT product family	Data encryption (restricted to export)

2.2 Heterogeneous networks

A group of interlinked computers and other devices is referred to as a network. When computers with the same type of communications structure are linked, we use the term homogeneous network.

The term heterogeneous network is used to denote a computer network in which computers intercommunicate with different communication architectures. Essential properties of computer networks are distances to be covered, the type transmission route, the utilization of public services and the type of protocols, i.e. the entire range of rules and regulations which must be observed for information transfer.

The most renowned networks supported by openFT are TCP/IP, NEA, ISO, SNA, X.21/X.25, ISDN. Not all network types are supported on all platforms.

Network management in heterogeneous networks are based on **SNMP** (Simple Network Management Protocol) in most cases.

The openFT products support the SNMP-based network management and thus underline their import in open networks.

The openFT products can use a variety of different transport systems with different transport protocols.

For an overview of the transport systems and protocols that permit the operation of openFT products, please refer to the relevant product data sheets.

openFT for z/OS supports the openFT protocol and the FTP protocol for transferring files.

2.2.1 openFT partners

openFT can perform file transfer and file management between partner systems which support the openFT protocols NEABD and NEABF in the application layers.

These partner systems are referred to below as openFT partners. openFT partners can run on mainframe platforms (BS2000/OSD, z/OS) and on open platforms (Unix systems, Windows systems).

Depending on the particular transport system software, a variety of transport protocols may be used:

– TCP/IP transport protocols
– SNA transport protocols

The range of functions is largely identical for a given openFT version across the different platforms, and any minor differences are the result of the operating system used.

These protocols, which were originally referred to as FTNEA protocols, have been opened, so there are now also products from other manufacturers that support these protocols.

2.2.2 FTP partners

Alongside openFT partners, it is also possible to address FTP servers.

If the FTP protocol is used then only communication via TCP/IP is possible. Furthermore, a number of special considerations apply when FTP servers are used compared to openFT partners. These are for the most part due to limitations in the FTP protocol:

- No restart is performed.
- Encryption is only possible for outbound requests to an FTP server that provides support for Secure FTP with the TLS protocol. This requires openFT-Crypt (openFT-CR delivery unit) to be installed.
- If encryption of the user data is required and the standard Secure FTP server does not provide encryption, the request is rejected. If encrypted transfer of the user data is required, the login data is also encrypted. If encryption of the user data is not required, the login data is only encrypted if the standard Secure FTP server provides this. No mutual authentication is carried out.
- Coded character sets are only supported locally; specifications for the partner system cannot be transported by the FTP protocol.
- When files with a record structure are transferred in binary format, the record structure is lost. The contents of the records are stored in the destination file as a byte stream.
- File attributes are not supported by the FTP protocol. This means that the modification date and maximum record length are not taken over for the destination file.
- If the *ftexec* command is issued to a mainframe over the FTP protocol, the *-t* option must be used. The *-b* option (default) is rejected in the remote system with a message indicating that the file structure is not supported.
- Follow-up processing is only possible on the local system or by specifying the FTAC profiles.
- The modification date cannot be taken over for the destination file. As a result, the modification date of the destination file is set to the transfer date. This is of particular importance when comparing file hierarchies.
- If an FTP server does not provide the information as to whether a symbolic link refers to a file or a directory when listing directories, the link is by default shown as a file in openFT Explorer (on Unix and Windows systems).

- The maximum record length of the send file is not passed to the receiving system. This has an impact when transferring files to a mainframe system such as BS2000/OSD or z/OS. In this case, the default maximum record length applies in the receiving system. If a record in the file exceeds this length, the request is cancelled with the message “File structure error” (FTR2210).
- The size of the send file is not passed to the receiving system. This has an impact when transferring files to a mainframe system such as BS2000/OSD or z/OS. The maximum file size is derived from the default value that is used by openFT for primary and secondary allocation and by the maximum number of file extents defined by the system, see section “BS2000/OSD files” on page 62 and section “z/OS files” on page 63. If a file exceeds this size, the request is cancelled with the message: “File gets no more space”.
- The 'do not overwrite' option (WRITE-MODE=*NEW-FILE) can have a different effect because this option cannot be passed to the responder, and the initiator must check whether the file already exists in the partner system. This has the following consequences:
  - It is possible for a request with the ’do not overwrite’ option (WRITE-MODE=*NEW-FILE) to overwrite a file that has been created by a third party in the period between the check being performed by the initiator and the actual transfer.
  - If 'overwrite' is specified in an FTAC profile (WRITE-MODE=*REPLACE), and if the file to be transferred does not yet exist, a request using this profile will still be executed, even if 'do not overwrite' (WRITE-MODE=*NEW-FILE) is set in the request.
- If you access password-protected mainframe files with a standard FTP client, e.g. in text format (C'password') or hexadecimal format (X'0A6F73'), you must append the password to the name of the remote file separated by a comma.

  Example

  ```
  put localfile remotefile,X'0A6F73'
  ```

Please note that the other openFT functions (preprocessing and postprocessing, FTAC, etc.) can only be used if openFT is used as the FTP server on the system, where preprocessing and postprocessing are to be performed.

Problems may also occur when addressing FTP servers which send an unexpected layout when listing directories.

2.3 Transferring files

The main function of openFT is to transfer files between two partner systems. To do this, you must issue a file transfer request in the local system. This request can be used either to send a file to a remote system or to fetch a file from a remote system to the local system. A partner system can also send files to your local system or fetch one from your local system.

Requests issued from you local system are referred to as **outbound requests** (sent from outside). Requests issued from the remote system are referred to as **inbound requests** (received from outside).

In a file transfer request, you can specify whether the file to be transferred is a text file or whether it contains unstructured or structured binary data. This determines the handling of the data during transmission; see the section “File conversion” on page 21. The so-called “transparent” file format plays a special role here: you can use this format to store BS2000 files with all their properties in the receive system without conversion. This is necessary, for example, when a Unix or Windows system is used to distributed BS2000 software.

Preprocessing, postprocessing and/or follow-up processing can be agreed for all file transfer requests to openFT partners. You may specify follow-up processing for successful and failed transfers both in the local system and in the remote system. For details of how to use the preprocessing. postprocessing and follow-up processing features, see the section “File transfer with preprocessing, postprocessing and follow-up processing” on page 33.

You should not process a file further until transfer is completed; otherwise, inconsistencies may result.

You may decide when openFT is to carry out your transfer request. Either immediately or at a particular time which you can specify. openFT always performs a synchronous request immediately. If a request is to be performed later, you must start an asynchronous request and specify the time of its execution.

Compressed transfer

When issuing a request, you may specify whether the file is to be transferred in a compressed form and the type of compression that is to be used (byte compression or zip compression).

Data compression can be used to:

– shorten transmission times
– reduce the load on the transmission paths and
– reduce data transmission costs.

2.3.1 Specifying the transfer start time

When you start a **synchronous request**, the file is transferred immediately. During the entire transmission period, a display on screen allows you to follow the progress of the file transfer and you have the advantage of knowing immediately whether or not the transfer was successful. You can use the result as decision criterion for further steps. If transfer failed because the partner was not available, for example, the file transfer is aborted and you can restart the request later.

In the case of an **asynchronous request**, openFT transfers the file either at the next possible time or at the time you specify. This allows the file transfer to be started at a time when the partner is available, or when transmission charges are particularly low. The request is stored in a request queue and you receive confirmation that the request has been accepted. Your system is thus immediately free for other tasks and you do not have to take care of executing the request. Thus, for example, if it is not possible to set up a connection for file transfer at a particular time, openFT re-attempts start of file transfer at defined intervals; even if a fault occurs during transfer, it is restarted automatically.

You can start several asynchronous requests. The requests are placed in a request queue until they are successfully executed, or cancelled by you or their maximum lifetime as set globally has been reached (see the section “Controlling the duration of a request” on page 28). You can use the request queue to obtain information on all request that have not yet been executed.

Requests issued by a remote system, i.e. inbound requests, are always executed as asynchronous requests in the local system by openFT.

2.3.2 Controlling the duration of a request

An asynchronous openFT request remains in the request queue until it is fully executed or explicitly deleted or until its lifetime, which can be set via an administration parameter, expires.

When issuing an asynchronous request, however, you may specify a time at which the request is to be deleted, or the file transfer is to be canceled (cancel timer). In this way, you can avoid tying up resources for partners who are temporarily unavailable, or when network problems are encountered.

2.3.3 Request queue

The request queue stores all asynchronous file transfer requests which have not yet been executed. You may display these on screen at any time. The information displayed will include:

– the transfer direction
– the operational status of the request
– the number of bytes already transferred
– the initiator of the request
– the local file name, for outbound requests also the remote file name.
– the partner system involved
– follow-up processing
– diagnostic information

The byte counter in the request queue is updated at regular intervals, so that you can keep up-to-date on the progress of file transfer.

You may delete requests change the order of the requests in the request queue (priority control).

For information on requests that have already been completed, use the logging function (see the section “Logging openFT operations - the logging function” on page 46.

Priority control

The requests are processed according to the FIFO principle (FIFO = First In First Out), i.e. the request issued first is processed first. Three priority classes (high/normal/low) are possible. You can control the processing of a request by:

– explicitly specifying the priority of a request
– changing the priority of a request in the request queue
– changing the queue of the request queue, i.e. placing requests at the start or end of a list of request with the same priority

Prioritization of partners

Partners can be prioritized in the partner list. This priority only applies to requests that have the same request priority, but are sent to partners with different partner priorities. Otherwise, the request priority overrides the partner priority.

The list below shows the sequence in which requests are processed if requests with different request and partner priorities are present.

Processing sequence	Request priority	Partner priority
1	high	high
2	high	normal
3	high	low
4	normal	high
5	normal	normal
6	normal	low
7	low	high
8	low	normal
9	low	low

2.3.4 Automatic restart

In the event of file transfer being interrupted for any reason, openFT provides for secure restart. This means that network problems, for example, present no difficulty to openFT, since openFT automatically continues transfer as soon as it becomes possible again.

The storage of the request in the request queue and the so-called restart points for the basis for automatic restart. These are the security points with which the two partner systems are synchronized at regular intervals during file transfer. If transfer is interrupted, it is continued as soon as possible starting at the last security point. You can therefore rest assured that not one single bit is lost and nothing is added during file transfer.

The fixed timing between security points ensures that no unnecessary security points are set for fast lines, and that the intervals are not too long for slow lines.

2.4 File management

In addition to file transfer, openFT offers the option of managing files in the remote and local and remote systems. You can perform file-management actions both with openFT statements and as processing within a file transfer request. It is expedient, for example, to formulate the necessary conditions for transfer or follow-up processing in the remote system prior to start of file transfer. This can be useful when creating file management requests prior to file transfer to the remote system, or when setting up conditions for follow-up processing, for example.

Furthermore, local or remote systems can be controlled from a Windows or Unix system via a user-friendly interface similar to the Windows standard, without the user having to be acquainted with the syntax of the remote system.

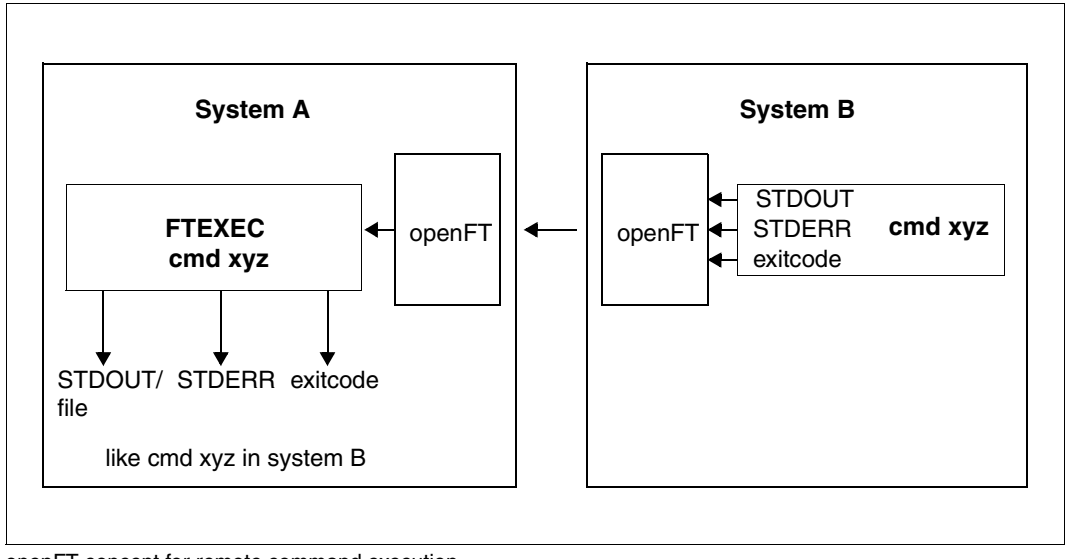
You can perform the following actions with via file management:

– rename files
– delete files
– query file attributes, e.g. the size of a file
– modify file attributes, e.g. access rights
– display directories
– create directories
– rename directories
– delete directories

2.5 Remote command execution

openFT for enables operating system commands to be executed on remote systems and can return the exit codes and outputs of such commands as if they were executed on the local system. This makes it possible to integrate remote commands transparently in local command procedures.

The following diagram clarifies the concept of remote command execution.



openFT concept for remote command execution

2.6 Automation

openFT provides job management functions such as file transfer with preprocessing, postprocessing and follow-up processing, the use of Monitor Job Variables in BS2000, and the use of file-transfer functions in dialog procedures and via program interfaces. Automation is also supported by the option for controlling the start time and lifetime of requests; see the corresponding sections. The creation of unique file names by using openFT variables makes it easier to design applications and reduces the amount of updating work to be done.

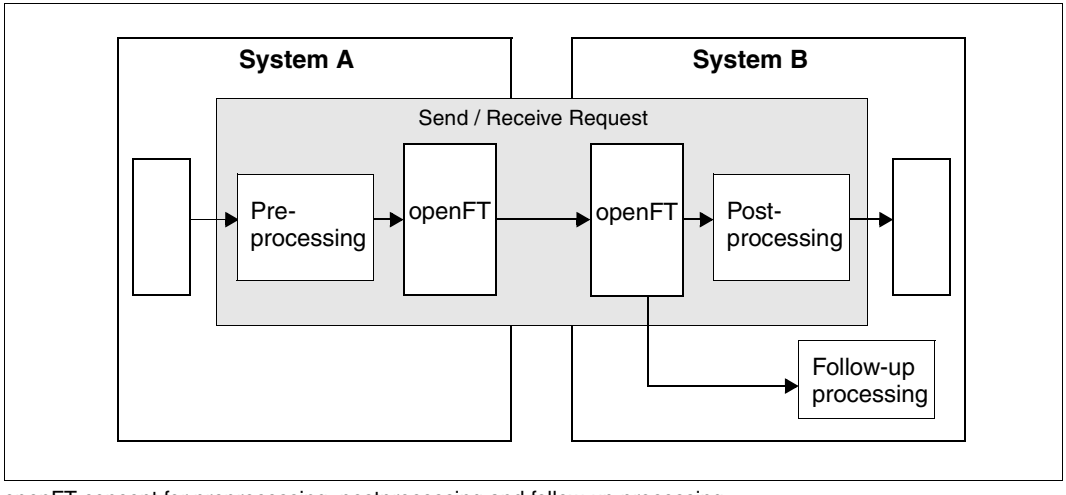
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing

For a file transfer, you can specify

– whether any preprocessing or postprocessing is to be done within a request. Preprocessing in the sending system and postprocessing in the receiving system are always possible and can also be combined within a request.

– whether any follow-up processing is to be performed after the file transfer. Follow-up processing can be defined for successful and unsuccessful file transfers both for the local and the remote system.

The following diagram clarifies the concept of a file transfer with preprocessing, postprocessing and follow-up processing.



openFT concept for preprocessing, postprocessing and follow-up processing

Pre- and postprocessing always take place within the openFT request, and follow-up processing always take place after the request.

In order to prevent system resources from being unnecessarily tied-up in a continuous processing loop, requests should be provided with a specified abort time if necessary.

2.6.1.1 Preprocessing

During preprocessing, you can, within a file transfer request, prepare the send data **before** the transfer using one or more commands. These could be operating system commands, program calls or procedure calls, in order to create or prepare the data before the transfer. The commands can, for example, extract information from a large data base (data base query), or prepare data (compress, encrypt), in order to subsequently pass it to openFT for file transfer.

2.6.1.2 Postprocessing

During postprocessing you can, within a file transfer request, process the received data using one or more commands **after** the actual transfer. To do this, you can execute commands, e.g. operating system commands, a program call or a procedure call. The command(s) can, for example, decode/uncompress data which has been encrypted or compressed using external routers.

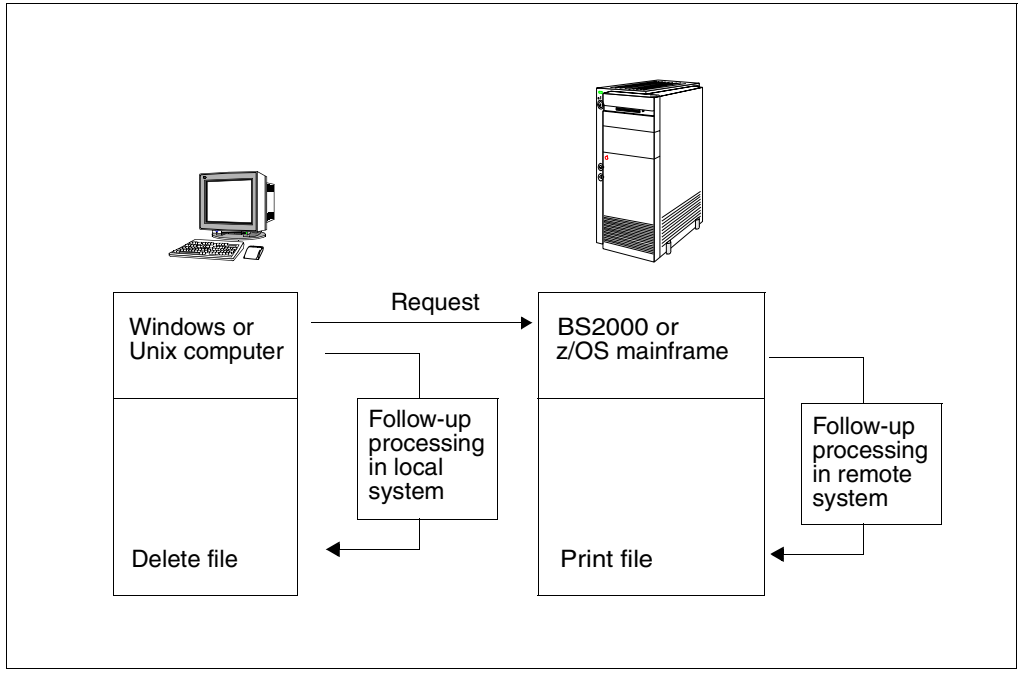
openFT requests with remote preprocessing or postprocessing can also be transferred by older versions of openFT or FT. It is important that a version of openFT that supports postprocessing is used in the remote system.

2.6.1.3 Follow-up processing

The "follow-up processing" option which is available in openFT enables you to execute sequences of statements or commands in the local and/or remote system depending on the positive or negative result of file transfer. If you specify follow-up processing for the remote system, you must observe the syntax of the operating system used on the remote system. When using commands, openFT provides variables which are replaced by the values in the file transfer request when the commands are executed.

Example

In the headquarters of a supermarket chain, there is a mainframe computer running BS2000 or z/OS. The branch office has Windows or Unix workstations. Every Saturday, the branch manager issues a request to transfer the file that contains a prepared list of the weekly sales. This file is transferred to the processor at the headquarters using openFT. The follow-up processing for the transfer request specifies that the file should be printed on the mainframe and then deleted from the branch computer if file transfer is successful.



File transfer with follow-up processing

2.6.2 Program interfaces

The program interface in openFT offers extensive automation capabilities. You can, for example, automate the issue of requests and request management in openFT, create your own user interfaces for openFT or integrate file transfer functions in other applications. In addition to the Java and C interface, an OCX interface is provided for Windows systems.

2.7 Further processing of openFT data

In order to permit openFT data (FTSHWLOG, FTSHWOPT, etc.) to be processed further by external procedures, openFT offers the so-called CSV (**C**haracter **S**eparated **V**alues) output format. In this format, each block of information is output to one line of text, with the individual items of information in an "output record" being separated by semicolons. The first line is a header and contains the names of the items of information, also separated by semicolons.

Such output could then be processed further by programs which support CSV formats (e.g. Microsoft Excel™ under Windows) and could hence be used, among other things, to easily implement an accounting system for the used resources (e.g. transfer requests).

2.8 Secure operation

Open networks, security during file transfer and data management are terms that need not be contradictory. openFT offers the following functions for secure operation are:

– individual settings for transfer and access rights with the FTAC function
– check of data integrity
– data encryption during the transfer
– logging function that can be enabled/disabled
– automatic encryption of the request description data
– Checking the communication partner using authentication

You can use these functions to make your system safe.

2.8.1 The FTAC function

With the FTAC function of openFT, you have all the options in your hand to make your system as secure as possible and as safe as it needs to be. FTAC stands for “File Transfer Access Control”.

FTAC offers the following protection mechanisms for your system:

– decoupling of FT transfer and login admissions
– access rights dependent on the partner systems
– user-specific access rights
– flexible access right levels
– recording of every authorization check
– simple application

2.8.1.1 Features of the FTAC function

For file transfer, a distinction is made between various functions. For access protection, the file transfer function being executed by the system is decisive. At first glance, there are only two such functions:

– sending a file and
– receiving a file.

Sending a file entails transmitting data from the system to be protected, while receiving a file involves the transfer of data into this system. However, for reasons of data security it is also important to know who requested a function in the system being protected. In FT terminology, this person is referred to as the initiator or submitter of the FT request.

Initiators can be divided into two groups:

– those in the system being protected (**outbound requests**)
– those in partner systems (**inbound requests**)

With this information, we can now make a distinction between four basic functions:

- **Outbound send**
- **Outbound receive**
- **Inbound send**
- **Inbound receive**

The possibility of processing transfer data (pre-, post-, and follow-up processing) during a file transfer should be considered an additional function. For FT requests submitted in the local system, no additional protection is necessary since anyone in the local system allowed to initiate FT requests already has access to the available resources. Processing in the remote system does not require any protective measures in the local system either. One function that does require protection in the local system is

- **Inbound processing**

which is initiated from a remote system.

Partner systems also have the option of using the file management functions to view directory or file attributes in their local system, to modify file attributes and to delete files and directories. This results in a further function:

- **Inbound file management**

File management, unlike the other functions, encompasses several different request options, which in turn are partially linked to the functions *inbound send* and *inbound receive*:

Inbound file management function	**Prerequisite**
Show file attributes	Inbound send permitted
Modify file attributes	Inbound receive **and** inbound file management permitted
Rename files	Inbound receive **and** inbound file management permitted
Delete files	Inbound receive permitted

The protection mechanisms offered by the FTAC function are primarily achieved through the use of admission sets and admission profiles.

2.8.1.2 Admission set

The admission set contains the basic specification of which file transfer functions are permissible. An admission set applies to exactly one login name. When access is attempted under this login name, FTAC checks whether the values set in the admission profile are complied. You can either restrict or extend the specification for the admission set using admission profiles or privileges respectively. If your security requirement is very high, we recommend that you block all inbound functions in your admission set, i.e. all possibilities of reaching your computer from the outside. You can then use the admission profile to permit one or more individual inbound functions for particular partners. In the admission set, the *outbound send* and *receive* functions assign transfer permissions to all partners under the relevant user ID.

You can view admission sets at any time and modify as required to meet your current needs.

Following installation of openFT the entries in the standard FT profile initially apply to all login names. The FTAC administrator must modify this standard FT profile after installation so that it provides the necessary protection for the majority of the login names. If individual login names require greater protection, the administrator can create specially adapted admission sets.

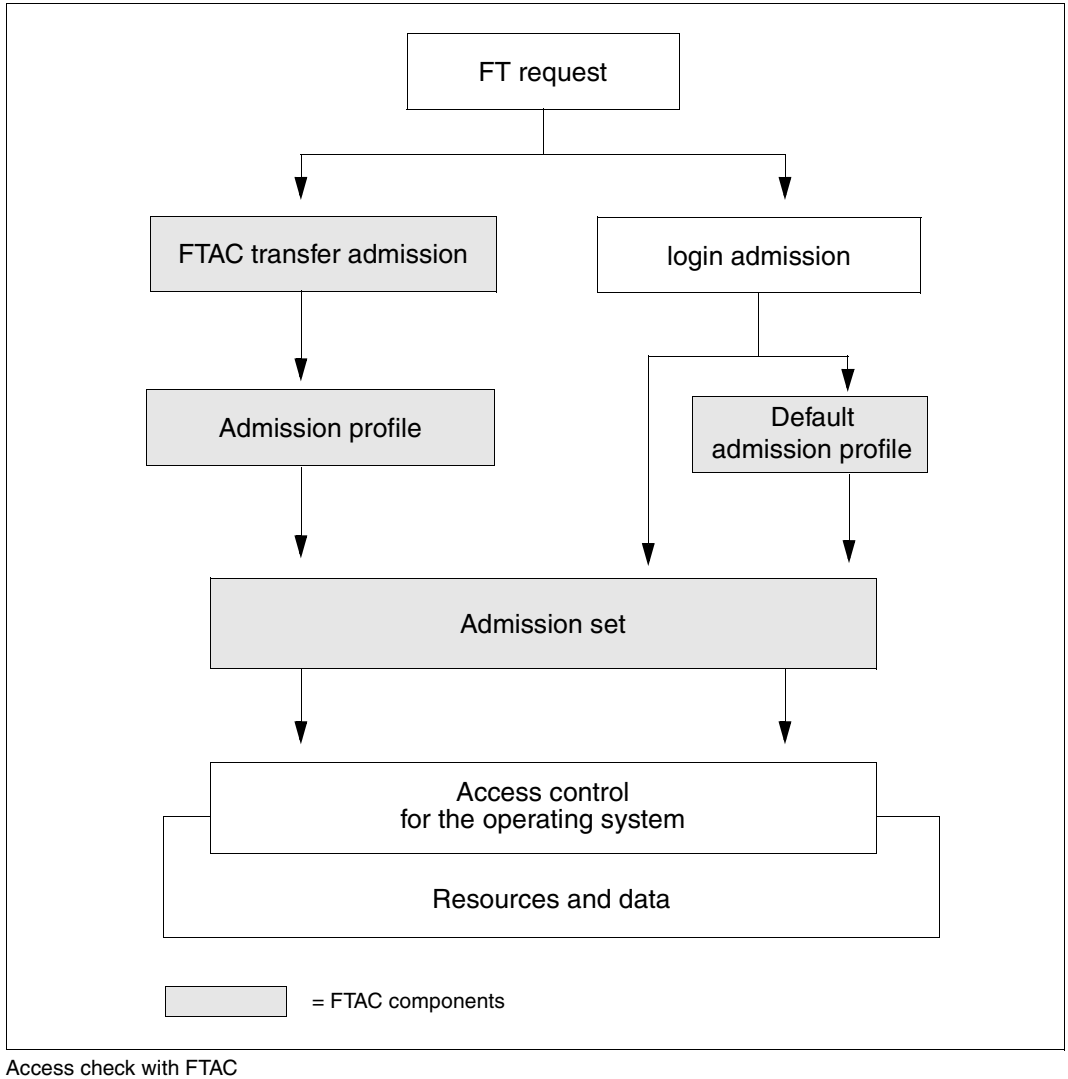
In addition, the FT administrator can assign security levels to the partner systems. When combined with the admission set settings, this makes it possible to prohibit or permit the use of the individual file transfer functions on a partner-specific basis.

2.8.1.3 FT profile (admission profile)

The FT profile (or admission profile) defines the **transfer admission** and the associated **access rights**. The transfer admission is the actual key to your processor. You should therefore treat the transfer admission with the same care as you look after a password. It must be specified in transfer requests instead of a login admission. The standard admission profile for a user ID is an exception. See page 43. Anyone who possesses this transfer admission does have file transfer access to your processor, but, unlike the Login admission, is not free do as he or she please. Which functions you permit are specified with the access rights for this transfer admission. In this way, you can control the conditions under which file are accessed or the follow-up processing commands which are permitted after file transfer. In the most extreme case, you can restrict access to your processor so much only on single profile is available providing access to only one file.

FTAC checks whether the entries in the request conflict with the entries in the FT profile for each file transfer request. If so, the file transfer request is rejected. In this case, only a general error message appears in the remote system.
This prevents the definition of the FT profile being established step-by-step on a trial and error basis. A log record which describes the cause of the error precisely is created in the local system.

The following diagram shows the sequences for admission checking with FTAC.



Access check with FTAC

An admission profile includes the following:

– a transfer admission. This transfer admission must be unique. If a request is to work with the FT profile, this transfer admission must be specified. FTAC only permits access rights for this request which are defined in the FT profile. In order to uniquely assign the responsibility for request, it is recommended that a transfer admission be assigned to exactly one person in precisely one partner system.

– if necessary, specification of the partner systems which may access this FT profile.

– Specification of the parameters that may be used in a request. In this way, the access rights are restricted for each person who uses this FT profile.

– If necessary, specification of whether and how long the FT profile is valid.

– A file name prefix. This prefix contains a part of the path name. The user of the profile can only navigate below this specified path name. For example, C:\Users\Hugo\ as a file name prefix on a Windows system means that the user of this profile can only access directories below the path C:\Users\Hugo\. The same principle applies on a Unix system if, for example, /home/hugo is specified as a file name prefix.

  On z/OS, for instance, a filename prefix is understood to be the "first-level qualifier" and where appropriate one or more further qualifiers, e.g. 'OPUSERS.HUGO.NEW.'.

  This prevents anyone with this profile to navigate within locked directories or from using the preprocessing function. Note, however, that it is also possible to specify a remote preprocessing command as the file name prefix, in which case, only the parameters for that command would then need to be specified in the request.

You can store various FT profiles.

You are always free to carry out the following operations on FT profiles:

– **Modify**
  and thus adapt the profile to current requirements.

– **Lock**
  In this case, a request with the locked profile is rejected on account of the invalid transfer admission. If you want to use the FT profile again, you must first unlock it.

– **Delete**
  You should limit the number of your FT profiles by deleting profiles which you no longer require.

– **Grant privilege** (system-dependent)
  In special cases, FT profiles can also utilize a function that has been locked in an admission set. In order to do this, the FT profile must be assigned a privilege by the FTAC administrator.

You may display information about your FT profile at any time.

Standard admission profile

You can set up a standard admission profile for each user ID.

This profile is only intended for certain use scenarios, such as when an FTAM partner has to specify the transfer admission in a fixed structure (user ID and password) for inbound access and you nevertheless wish to specify certain settings, such as a filename prefix.

Unlike a normal profile, a standard admission profile has no FTAC transfer admission, because access is controlled implicitly using the user ID and password. On the other hand, this profile allows most of the normal parameters to be set, such as the permitted FT functions, a filename prefix or the write mode. You cannot set the expiry period, whether or not the profile is locked and whether the profile is private or public.

A standard admission profile must be set up explicitly and a maximum of one standard admission profile can be set up for each user ID.

2.8.1.4 Effects of an admission profile

The following table contains possible restrictions to the access rights in an FT profile in the left-hand column, and the entries for the file transfer request required for the partner system in the right-hand column. Some differences apply to a standard admission profile. See above.

Entry in the FT profile	Entry in the file transfer request
Transfer admission	The transfer admission addresses the admission profile. If the user ID and password are specified, it is only possible to address the standard admission profile of the user, if this has been defined.
Transfer direction restricted	The parameter specified must be the opposite of the entry in the FT profile. If the profile contains transfer direction “From Partner”, the remote system may only send data to the local system; with “To partner”, it is only possible to transfer files to the remote system. In contrast, only read access is permitted in the local system.
Partner systems specified	The request can only be issued by the partner systems entered in the profile.
File name specified	The file name must be omitted in the request. If it is a mandatory parameter in the partner systems’s file transfer product, it must be assigned the value “*not-specified” (e.g. BS2000/OSD).
Prefix for the file name specified	Only part of the file name which is not is present in the request. FTAC supplements this entry with the prefix defined in the profile to obtain the complete file name. The specification of absolute file names, or exiting a directory with “..” is prohibited by FTAC.
rocessing prohibited	No processing may be requested for your processor.
Processing specified	No processing may be requested for your processor.

Entry in the FT profile	Entry in the file transfer request
Prefix/suffix for follow-up processing specified	Only the part of the follow-up processing defined in the profile may be specified in the request. FTAC supplements this entry to produce the complete follow-up processing command. If no follow-up processing is specified in the request, none is carried out.
Write mode restriction	The request is executed only if it complies with this write mode.
Force or forbid encryption	The request will only be carried out if it corresponds to the presets in the admission profile.

Migrating admissions

The FTAC administrator can store both complete admissions as well as individual admission records and profiles in a file (migration). You can then take from the file as required.

2.8.1.5 FTAC administrator

openFT offers the FTAC function for platforms ranging from PC to mainframe. On some stand-alone system the user is responsible for all administrative tasks, whereas large multi-user systems, such as mainframes, offer a multitude of administrative tasks as a centralized service. The FTAC function offers options for these “administration scenarios” by giving, for example, the user of openFT for BS2000/OSD, z/OS, Windows systems or Unix systems the possibility to rely on his or her FTAC administrator. The FTAC administrator, who is not necessarily identical to the FT administrator, also specifies the security framework for his or her system in the form of a standard admission set which is applicable to all users. The individual user then has the option of customizing the security mechanism set by the administrator to meet individual requirements, or to accept the setting made by the FTAC administrator as the lowest security level for his or her system.

2.8.2 Encryption for file transfer requests

When connecting to openFT partners that support the AES algorithm (e.g. openFT V8.0 and higher), then RSA/AES encryption algorithm is used for the request description data and the content of the transferred file.

To do this, openFT as V12.0 uses a 2048-bit RSA key by default. Alternatively, a 1024-bit or 768-bit RSA key can be used. The FT administrator must set this in the operating parameters. In the case of connections with older versions, encryption is negotiated downwards if necessary, i.e. an RSA of a length that is available in the older version is ised or, if RSA keys are not supported, DES encryption is employed.

For encryption in file transfer requests, a distinction must be made between request description data and user data.

The encryption of the user data is only possible if this function has been enabled with the corresponding module (openFT-CR). This product is subject to export restrictions.

The encryption of user data is only available for data transfer with openFT partners.

Encryption of request description data

Request description data contain security-relevant information, such as addresses and passwords which give access permissions. The encryption of request description data is agreed automatically between the partner systems when a connection is set up, provided both partners support encryption. Otherwise the request description data is transferred unencrypted.

Encryption of the content of the file to be transferred

Stricter requirements for data security are satisfied by the option of encrypting user data as well. With openFT you can

- purposely request an encrypted transfer of your user data during outbound requests
- force or forbid encryption of user data using an admission profile during inbound requests.

In addition, the FT administrator can force the general use of data encryption for inbound and outbound requests by making the appropriate settings in the operating parameters.

If your FT partner does not offer this capability, or it does not adhere to the presets in the admission profile, then the request will be denied.

Please note that the overhead required for data encryption produces a trade-off with system performance at the partner.

It is possible to control encryption in the admission profile:

- Encryption can be explicitly forced, for example, for requests requiring an especially high degree of security. Requests with unencrypted user data will be denied.
- Encryption can be explicitly forbidden, for example, for requests requiring a lesser degree of security, where performance is key. Requests with encrypted user data will be denied.

The mechanism for active encryption of user data is a separate delivery unit and must be released explicitly due to legal requirements.

2.8.3 Logging openFT operations - the logging function

Prevention of unauthorized access and protection of data inventories is just one security aspect. The complete documentation of the access check and the file transfer requests also puts you in a position to check your security network at any time and detect any leak. The logging function of openFT is the most suitable tool for doing this. It is activated as default and logs all information relating to file transfer requests, irrespective of whether the initiative lies in the local or remote system and whether the transfer was successful or not. The **log records** are written into the corresponding file. The scope of logging can be set as appropriate.

The logging function also serves as a basis for detecting break-in attempts. In addition, it may be used to obtain and evaluate performance data (see also the section “Further processing of openFT data” on page 37).

Log records

If your local system is protected by FTAC, FTAC first checks all accesses to your system and logs the result in an **FTAC log record**. If the access check is negative, FTAC already rejects the request. If the access check is positive, the following applies:

- In the case of a file transfer request (and if the request materializes), an **FT log record** is subsequently written indicating whether the request was executed successfully or why it was cancelled. This means that there can be two log records for one transfer request.
- In the case of a remote administration request, an **ADM log record** is written indicating whether the request was executed successfully or why it was cancelled.

You may display log records relating to your login name at any time, either in abbreviated form or with all data. You may also display only particular log records. e.g. all log records for a certain partner system. The log record provides the following information:

– Type of log record (FT, FTAC or ADM)
– Date and time when the log record was written
– A reason code which informs about the success or failure of the request
– Name of the partner system
– Direction of file transfer
– Identification of the initiator for outbound
– Name of the file in the local system

Log records of other login names can only be viewed by the administrator.

Offline logging

The FT administrator can switch the log file during system operation. Following the switchover, new log records are written to a new log file. The previous log file remains available as an offline log file. You can continue to view the log records for your user ID using the tools available in openFT.

Logging request with preprocessing / postprocessing

For security reasons, only the first 32 characters of a preprocessing or postprocessing command are recorded in the log record. The user can influence which command parameters will appear in the log file by arranging the call parameters accordingly or by entering spaces in the list of parameters.

Specifying the scope of logging

the FT administrator has the following selection options for the FT log record:

– never log
– log only errored file transfer requests
– log all file transfer requests

All file transfer requests are logged as default.

As FTAC administrator, you have the following selection options for the FTAC log record:

– log only rejected FTAC access checks
– log only modified file management requests and rejected FTAC access checks
– log all FTAC access checks

All FTAC access checks are logged as default.

The FT administrator can choose between the following options for the ADM log record:

– never write a log record
– only log failed remote administration requests
– only log remote administration requests that modify data
– log all remote administration requests

By default, all remote administration requests are logged.

Saving and deleting log records

Only the FT administrator, the FTAC administrator and the ADM administrator are permitted to delete a log record or log file. Log records should be saved at regular intervals (ideally using a cyclical job). During this, the output of the FTSHWLOG command, not the active log file itself, should be saved. Switching the log file makes it possible to save the current log records in an offline log file. This offline log file can then be backed up by the FT administrator.

The benefit of this is, first, that the log records provide a complete record of FT operations which can be maintained for long periods, and second, that the log file does not assume unnecessarily large proportions, which saves CPU time when accessing the records.

2.8.4 Authentication

If data requiring an extremely high degree of security is to be transferred, it is important to subject the respective partner system to a reliable identity check (“authentication”) before the transfer. The two openFT instances engaged in the transfer can perform mutual checks on one another, using cryptographic resources to determine whether they are connected to the “correct” partner instance.

To this end, openFT supports an addressing and authentication concept that is based on the addressing of openFT instances via network-wide, unique IDs and the exchange of partner-specific key information.

Instance identification

Each openFT instance that works using authentication, must be assigned a network-wide, unique instance identification (instance ID). This is a name, up to 64 characters long, which, as a rule, should correspond to the DNS name of the openFT instance. The unique instance ID must not be case-sensitive. The FT administrator defines these IDs for the local system using an operational parameter. Instance IDs of partner systems are stored in the partner list. openFT administers the resources assigned to these partners, such as request waiting queues and cryptographic keys, with the aid of the instance IDs of the partner systems.

Key administration

The FT administrator can prepare a maximum of three RSA key pair sets, each of which consists of a private and a public key, for each local openFT instance. The public keys are stored under the following name at the following location:

<openft qualifier>.<inst>.SYSPKF.R.<key reference>.L<key length> in the openFT instance.

The key reference is a numerical designator for the version of the key pairthe key length is currently 768 bits, 1024 bits or 2048 bits. The public key files are text files that are created in the character code of the given operating system, i.e. as standard:

– BS2000/OSD: value of the system variable HOSTCODE
– z/OS: IBM1047
– Unix systems: ISO8859-1
– Windows systems: CP1252

In order that one's own openFT instance can be authenticated in the partner system, the appropriate public key must be made available to the partner system. This should take place via a secure path, for example by

– distribution by cryptographically secure e-mail
– distribution on a CD (by courier or registered mail)
– distribution via a central, openFT file server, for which you have a public key.

If the key files between Windows or Unix systems and BS2000 or z/OS are exchanged, you must ensure that these files are re-coded (e.g. by transferring them as text files via openFT).

The FT administrator can use the command FTIMPKEY to import a partner system's public key.

2.9 Using openFT in a Sysplex composite

In openFT you can simultaneously execute more than one openFT instance on a single host. This allows you to switch to the openFT functionality on a different computer that is already running openFT when your computer fails.

openFT commands that can be called during preprocessing, postprocessing or follow-up processing execute in the same instance as the request that initiated the preprocessing, postprocessing or follow-up processing.

To set up in which Instance openFT commands should run, you must explicitly assign the corresponding files (see section “Setting an openFT instance” on page 148).

Furthermore, you can output information on the instances with the FTSHWINS command.

You will find a detailed description of the commands in the command chapter.

3 File transfer and file management

File transfer with openFT is initiated by a file transfer request. In the file transfer request, you make entries to specify the partner system, the transfer direction, the file name and file properties. Given the variety of hardware and software platforms supported, the values specified are subject to various different conventions applicable to the operating systems involved in file transfer. Which files can be transferred between two computers depends on whether the file transfer partners are running identical operating systems (homogeneous link), or different operating systems (heterogeneous link). The file management offered by openFT allows you to delete, rename files, or change file attributes before or after file transmission or even without file transfer.

The use of the FTAC functionality offers you not only security benefits, but also allows you to make your file transfer operating system independent (see the section “Features of the FTAC function” on page 38), provided the appropriate FTAC settings exist on the processors involved in the file transfer.

Entries for file transfer requests

The following sections give you an overview of the entries you have to make for a file transfer request. They are divided into a local, a remote and an optional part. In the local part, you specify the local file name, if necessary, with the directory name and the file passwords. In the remote part, you define the remote file name, the partner computer and the access to this processor (login name and, if antecessor, the account number and password or transfer admission). In the optional part, you have the option of specifying transfer modalities, such as file types, and follow-up processing requests, for example.

3.1 File names

The description below provides an overview of the system-specific conventions for entering file names, regardless of whether a local or remote file name is involved. By using the FTAC functionality with an appropriate definition in the FT profile, you can avoid having to enter all or part of the file name (see the section “FT profile (admission profile)” on page 40). In other words, the parts of the file name defined in the FT profile need not specified in the file transfer request again.

3.1.1 Unique file names for receive files

One of the important applications of openFT products is to transfer a file to a target system with automatic follow-up processing of the received file. In many cases, the receive file is actually only an intermediate product of the processing involved. In order to prevent potential conflicts with concurrently running requests in such cases, the metacharacter string %UNIQUE or %unique can be specified in the receive file name as to instruct openFT to create a unique file name. openFT replaces %UNIQUE or %unique with a string of variable length.

This string is 14 characters long in Unix systems, 18 characters long in Windows systems, 22 characters long in BS2000 systems and 15 or 8 characters long (for libraries) in z/OS systems. If the receiving system is a Unix or Windows system, a suffix may follow %unique or %UNIQUE separated by a dot, e.g. "file1%unique.txt". This suffix must not contain any dot.

In z/OS, openFT converts %UNIQUE into a string of the form xxxxxxxx.D<inst>, where xxxxxxxx stands for a randomly generated string, D for a letter and <inst> for the name of the openFT instance that is currently set.

%UNIQUE must be a separate part of the file name (separated by .) for the records of the normal z/OS file system. This can be followed by a maximum of one further name component.

%UNIQUE can also be specified for a member name of a PO or PDSE dataset. However, in this case only the name part to the left of the period (xxxxxxxx) is replaced.

In the case of openEdition files, %UNIQUE is expanded if:

– it is at the end of the file name.

– it is followed by a string which contains neither a percentage sign nor a period (unless the period immediately follows %UNIQUE).

%UNIQUE is recognized and correctly converted to:

– receive file names, on initiation from the remote system with WRITE-MODE = *REPLACE-FILE or *NEW-FILE

– receive file names of the NCOPY command with WRITE-MODE = *REPLACE-FILE or *NEW-FILE

– file names in FTCREPRF or FTMODPRF to preset the receive file name in FTAC profiles

Note that in log records, result logs, the output of the NSTATUS command, and in the messages FTR0340 and FTR0341, the file name is displayed using the values that have already been set for %UNIQUE.

The generated file name can be symbolically referenced in the follow-up processing via the metacharacter string %FILENAME, %FILN or %FILX.

3.1.2 BS2000/OSD file names

Format for BS2000 (DMS)	Meaning
:cat:$user.filename	cat Optional specification of catalog ID; Available characters restricted to A…Z and 0…9; max. 4 characters; must be enclosed in colons; Preset is the catalog ID assigned to the login name in the entry in the user catalog.
	user Optional specification of login name; Available characters A…Z, 0…9, $, #, @; max. 8 characters; must not start with a digit; $ and the dot must be entered; Preset is the catalog login name under which the file is accessed.
	filename File name can be split up into several subnames: $name_1$[.$name_2$[...]] $name_i$ contains no blanks and must start or end with a hyphen; Character set is A…Z, 0…9, $, #, @. File name can be up to 41 characters long, must not start with $ and must contain at least one character in the range A…Z.
:cat:$user.group (gen-no)	cat see above user see above group Name of a file generation group For character set see filename, brackets must be specified max. length 41 characters. (gen-no) (*abs) absolute generation number (1..9999); * and brackets must be specified. (+/-rel) relative generation number (0..99); Signs and brackets must be specified.

Format for BS2000 (DMS)	Meaning
:cat:$user. lib/typ/element	cat see above user. see above lib Library name; the rules for BS2000 DMS file names apply.
	typ Element type; Alphanumeric name, 1 - 8 characters in length.
	element Element name; The rules for LMS element names apply; element can be up to 64 characters in length, must not begin with $, and must include at least one character from A…Z.

In the remote BS2000 operands for the POSIX file names, the POSIX file name must be specified as a C string (graphic string) (i.e. enclosed in quotation marks). This is necessary in order to distinguish between uppercase and lowercase in POSIX file names.

Format for BS2000 (POSIX)	Meaning
posix file name	Character string up to 255 characters long. Comprises either one or two dots, or alphanumeric characters and special characters; special characters must be canceled with \. The character / is not permitted. Must be enclosed in quotation marks if alternative data types are permissible, separators are used or the first character is ? or !. The POSIX file name must be prefixed with a POSIX path name.
posix path name	Input format: [./]$part_1$/.../$part_n$ where $part_n$ is a posix file name; up to 1023 characters; must be enclosed in quotation marks if alternative data types are permissible, separators are used or the first character is ? or !. The POSIX path name must begin with / or ./, or consist of at least / or ./.

3.1.3 File names in Unix systems

Up to 512 characters, where a distinction is made between uppercase and lowercase. It is recommended that the following characters be avoided in file names:
- ? @ # $ ^ & * ( ) ‘ [ ] \ | ; " < >.

3.1.4 Windows file names

File name here refers to the complete pathname.

Up to 256 characters. The following characters must not be used:
| * ? " < >.

No network drives can be specified for remote file names, either when fetching or sending files. Instead, you can specify UNC names.

UNC names

UNC names (**U**niversal **N**aming **C**onvention) are addresses of shared resources in a computer network. They have the following format:

```
\\hostname\sharename\path\file
```

Either the host name or the IP address, for example, can be specified for *hostname*:

```
\\host1\dispatch\catalogs\winterissue.pdf
```

or

```
\\172.30.88.14\dispatch\catalogs\winterissue.pdf
```

3.1.5 z/OS file names

Format for z/OS	Meaning
':S:first-qual>.filename' or :S:filename	Specification for PS dataset :S: prefix for identifying a PS data set (no restrictions) first-qual “first level qualifier” Specification of login name; Available characters: A…Z, 0…9, $, #, @; max. 7 characters; must not start with a digit or alias name (max. 8 characters) filename partially qualified file name can be split up into several subnames using dots: $name_1$[.$name_2$[...]] $name_i$ is up to 8 characters long; available characters: A…Z, 0…9, $, #, @; must not start with a digit The partially qualified file name can be up to 36 characters long Fully qualified name The fully qualified file name (first-qual.filename) can be up to 44 characters long.
':S:first-qual. gengroup.Gmmmm.Vnn' or :S:gen-group.Gmmmm.Vnn	Specification for absolute file generation :S: prefix for identifying a PS data set (no restrictions) first-qual See “Specification for PS dataset” for syntax gen-group See filename in “Specification for PS dataset” for syntax Exception: partially qualified file name, up to 27 characters; fully qualified file name up to 35 characters Gmmmm.Vnn absolute file generation mmmm absolute generation number (0000 - 9999) nn version number (00 - 99)

Format for z/OS	Meaning
':S:first-qual. gen-group(rel-gen-no)' or :S:gen-group(rel-gen-no)	Specification for relative file generation :S: prefix for identifying a PS data set (no restrictions) first-qual See “Specification for PS dataset” for syntax gen-group See gen-group in “Specification for absolute file generation” for syntax rel-gen-no relative generation number 0 = current generation +/-m = 1 - 99 for partially qualified specification (without first-qual and quotation marks) 1 - 255 for fully qualified specification (with first-qual and quotation marks)
':prefix':first-qual. filename(membername)' or :prefix:filename (membername)	Specification for PO or PDSE member :prefix: prefix for identifying the file organization (no restrictions); can have the following values: :O: for PO :E: for PDSE :L: for PO or PDSE first-qual Syntax see “Specification for PS dataset” filename Partially qualified file name of PO or PDSE dataset Syntax see filename in “Specification for PS dataset” membername Name of PO or PDSE member max. 8 characters long, available characters: A…Z, 0…9, $, #, @; must not start with a digit
“:V:first-qual.filename” or :V:filename	Specification for VSAM file of type "entry-sequenced" :V: Optional prefix for designation of a VSAM file of "entry-sequenced" first-qual Syntax see "Specification for PS data set" filename Partially-qualified file name of VSAM file Syntax see filename in "Specification for PS data set"

Format for z/OS	Meaning
':prefix: first-qual.filename' or :prefix:filename	Specification for a complete PO or PDSE data set :prefix: prefix for identifying the file organization (no restrictions); can have the following values: :O: for PO :E: for PDSE :L: for PO or PDSE first-qual See "Specification for PS data set" for syntax filename partially-qualified file name of PO or PDSE data set See filename in "Specification for PS data set" for syntax Exception: maximum length of partially-qualified file name is 34 characters, fully-qualified file name is 42 characters. Thus the maximum permitted file name length is, for both partly and fully qualified specifications, 2 characters shorter than for a PS data set. This is because the name of a temporary data set required to transfer a complete PO or PDSE data set is formed by adding ".U" (see section “Transferring a PO or PDSE data set” on page 81).

Access to files of the z/OS Unix System Services (openEdition files) is supported as of openFT V10 for z/OS. The file names comply with the POSIX conventions.

Format with z/OS	Meaning
filename	Components of an openEdition filename. String up to 255 characters in length. This comprises either one or two periods or alphanumeric characters and special characters. The character / is not permitted.
pathname	openEdition file name Input format: [.]/[$part_1$/.../$part_n$] where $part_n$ is a POSIX file name; up to 512 characters. If the name starts with /, it is interpreted as an absolute path name. If the name starts with ./, it is a “relative” path name and is relative to the directory for the user ID, e.g. /u/userid in lowercase characters/.

The structure of the file name is used in openFT for z/OS to determine the type of the send file or receive file (see also section “z/OS files” on page 63).

With the **send file**, openFT checks whether the structure of the specified file name matches the file type of the send file; if it does not, the transfer job is rejected.

If, for example, the PS file ABC is sent, the file name can be specified as ABC or :S:ABC; if :V:ABC is specified, the job is rejected. (In fully-qualified name specifications, the name must be specified accordingly.) If member GHI of PO data set DEF is to be sent, the file name can be specified as DEF(GHI), :L:DEF(GHI) or :O:DEF(GHI); if it is specified as :V:DEF(GHI), the job is rejected. (In fully-qualified name specifications, the name must be specified accordingly.)

With the **receive file**, the following cases are possible:

– The receive file does not yet exist. openFT then creates a receive file of the type determined by the structure of the file name. If the file name allows several interpretations, the precise file type is determined using specifications made by the FT administrator or default values (for more information, see section “Attributes of receive files” on page 66).

– The receive file already exists and is to be overwritten or modified. openFT then checks, as with the send file, whether the structure of the specified file name matches the file type of the existing receive file; otherwise, the transfer job is rejected.

3.2 File passwords

If a password applies to a file that is accessed with openFT is password-protected, the password must be entered. In Windows and Unix systems, there are no file passwords.

System	File password
BS2000	1 - 4 character C string (graphic string) or 1 - 8 character X string (octet string) or integer string between 2147483648 and 2147483647
z/OS	1 - 8 alphanumeric characters

3.3 File types

Depending on their file type and the operating system from which they originate, files that can be transferred have different properties, which must be considered during the transfer.

3.3.1 BS2000/OSD files

In accordance with the different file structures, a distinction is made between the following BS000 file types:

- Cataloged files
  - DMS files (these include SAM, ISAM, and PAM files, PLAM libraries and cataloged generations of a file generation group)
  - POSIX files
- Elements of a cataloged PLAM library
  - Printable or user-definable elements of type D, J, M, S and possibly X
  - Elements with BS2000-specific binary code of type C, L, R and possibly X

In order to be able to transfer POSIX files using openFT, POSIX must be started. The POSIX file system essentially corresponds to the layout and structure of the Unix file system.

The following overview shows the relationship between file name syntax and file type in BS2000.

File name syntax	File type
Starts with $userid or :catid:$userid and does not contain '/'	DMS file, fully qualified
Starts neither with '/' nor with './' nor with $userid nor with :catid:$userid and does not contain '/'	DMS file path relative to transfer admission
Starts with '/'	POSIX file, fully qualified
Starts with './'	POSIX file, path relative to transfer admission
Starts with $userid or :catid:$userid and contains at least one '/'	Name of a PLAM element, fully qualified
Starts neither with '/' nor with './' nor with $userid nor with :catid:$userid but contains at least one '/'	Name of a PLAM element, path relative to transfer admission

BS2000 files may be located either on common disks or on private disks. For processing of files on private disks, the files must be cataloged and the private disks must be properly connected to the system.

3.3.2 z/OS files

openFT for z/OS can transfer the following types of files:

- PS datasets including absolute and relative file generations
- Members of PO and PDSE datasets (with the exception of object modules and programs)
- VSAM files of type “entry-sequenced”
- openEdition files (files belonging to the z/OS Unix Systems Services)
- Migrated files, i.e. files swapped out with HSM. See also the section “Migrated files” on page 95.

The transfer of these files is performed sequentially. The files can be transferred homogeneously between two z/OS systems or heterogeneously with a non-z/OS system or a non-z/OS system. For homogeneous file transfer, all file types can be mapped to one another. Between z/OS and other platforms (heterogeneous link) it is possible to transfer files if the remote system also supports sequential files. WithBS2000/OSD systems, for example, SAM files and PLAM elements of the appropriate type can be exchanged.

The transfer of complete PO and PDSE datasets can only take place between two z/OS systems.

z/OS files may be located either on common disks or on private disks. For processing of files on private disks, the files must be cataloged and private disks must be properly connected to the system. For the processing of files on private media, the precondition is that the files are cataloged and that the private data medium has been properly connected to the system.

Please note that at the command interface, a c string must also be entered in the form C’...’ as otherwise (without the C) openFT would try to interpret the string as a fully qualified z/OS file name.

Primary and secondary allocation

When openFT receive files are created in z/OS, the primary allocation approximately corresponds to the (possibly estimated) size of the send file (at least 42 kilobytes, however) plus 128 kilobytes (DEFFSIZE/20). The secondary allocation approximately corresponds to a quarter of the size of the send file plus 512 kilobytes (DEFFSIZE/5). DEFFSIZE is a constant that is set to 2621440 by default. It can be modified by making an appropriate entry in the PARM member of the parameter library. See the System Administrator Guide "openFT for z/OS - Installation and Administration".

If a PO/PDSE file is created by generating a member, the primary allocation is twice the size of the send file (at least 42 kilobytes, however) plus 256 kilobytes (DEFFSIZE/10). The secondary allocation is slightly less than twice the size of the primary allocation.

If the size of the send file is unknown to openFT internally (e.g. in the case of a file transfer with preprocessing and/or preprocessing using the FTEXEC command respectively), or if the size of the send file is not passed to the z/OS receiving system with the protocol used (as is the case, for example, with the FTP protocol), the primary allocation for the receive file in z/OS is 256 kilobytes (DEFFSIZE/10) and the secondary allocation is 2560 kilobytes (DEFFSIZE).

In the case of very large files, it is not always possible to reserve the entire space with a primary allocation, and there are also restrictions for secondary allocations. These limits depend partly on the hardware properties of the disks (a maximum of 65535 tracks per file on a volume) and partly on the current disk occupancy (in the case of multivolumes). For this reason, it is possible to restrict the maximum size of an allocation (both primary and secondary) to a maximum value MAXALLOC. See the System Administrator Guide "openFT for z/OS - Installation and Administration". If the allocations calculated using the method described above do not exceed this threshold, MAXALLOC is of no significance.

Encoding

In z/OS systems, the content of text files is coded in EBCDIC. However, the conventional IBM EBCDIC variants differ from EBCDIC.DF.04; in particular, this affects language-specific special characters (e.g. "ä", "ö", "ü") and other special characters (e.g. "[", "]", "{", "}") which may be located at different positions of the code table in the different EBCDIC variants. openFT provides a range of character sets (code tables), and if necessary, a specific character set can be assigned to each file; see also section “Transferring 7-bit, 8-bit and Unicode files” on page 96. Code conversion is performed by openFT, for instance between EBCDIC.DF.04 and IBM1047. This means that the FT administrator does not need to create any code conversion tables. It is, however, possible to set up your own code tables if the character set you require is not provided by openFT itself or by one of the supplied code tables.

A complete PO/PDSE data set is not converted if openFT as of V10 is used in the sending system. If this is not the case, steps should be taken to make sure that no conversion is carried out, otherwise it is possible that the control information in the destination file may no longer be correct.

The following files cannot be transferred by openFT:

– Files with the attribute “unmovable” (data organization PSU)

3.3.2.1 Files that can be transferred

The following overview shows what types of file you can transfer from or to a remote FT system:

Local openFT for z/OS	Remote FT system	Type of file
PO data set PDSE data set	openFT for z/OS	PO data set PDSE data set
PS data set PO member PDSE member VSAM-ES data set openEdition file	openFT for z/OS	PS data set PO member PDSE member VSAM-ES data set openEdition file
	openFT for B2000	SAM file LMS element POSIX file
	other FT system	sequential file

3.3.2.2 Volumes

openFT can transfer all files located on public or private direct-access volumes ("direct access storage devices", DASD). The following conditions apply to processing files:

– the user must have cataloged the files in the system catalog before the transfer.
– the user must have mounted the private volume before submitting the file transfer request.

openFT can read any multivolume files as send files. The following should be noted when writing to multivolume files:

– PO data sets cannot be written to multivolume files.
– Data sets with more than 20 volumes cannot be extended.

3.3.2.3 Attributes of receive files

This section describes the attributes of **new** receive files created by openFT (indicated by the entry "N" in the table on the next page).

The attributes described here also apply to receive files with which openFT **overwrites** existing files (indicated by the entries "O" and "D + N" in the table on the next page).

The following points also apply:

If, during transfer, the receive file is to be retained, i.e. you only want to **extend** the receive file (entry "E" in the table below), the attributes of the existing file are retained. In this case, transfer only takes place if the format of the new data is compatible with the attributes of the existing file. Otherwise, the transfer request is rejected with an error message or is canceled.

In all cases where a receive file is to be extended, note that the z/OS Data Management System permits a total of only 16 "extents" for PS and PO data sets. In the case of PDSE data sets, up to 123 extents are possible. If this is insufficient for the extension of the receive file, the file transfer is rejected with an error message or is canceled.

The Data Management System also issues an error message.

There are three factors which determine whether openFT creates a new receive file, overwrites an existing file or extends an existing file:

– the value of the WRITE-MODE operand in the NCOPY command,
– the type of the receive file, and
– whether or not a file with the same name already exists.

The following table shows which combinations of these factors cause a new file to be created or an existing file to be overwritten or extended for the purpose of storing the data transferred. The procedure always applies to the **entire file** (or to the PO/PDSE data set if a PO/PDSE member is being transferred).

Type of receive file	Existing file with the same name	Value of the WRITE-MODE operand		
		*NEW	*REPLACE	*EXTEND
PS data set	no	N	N	N
	yes	---	Ü	E
VSAM file	no	N	N	N
	yes	---	L + N	E
openEdition file	no	N	N	N
	yes	---	L + N	E
Library transfer:				
PO or PDSE data set	no	N	N	---
	yes	---	L + N	---
Member transfer:				
PO or PDSE data set does not exist		N	N	N
PO or PDSE data set exists, member does not exist		E	E	E
PO or PDSE data set exists, member exists		---	E	E

Explanation:

N A new receive file is created.

D + N The file is deleted (thus releasing the disk storage occupied by it) and a new file is created on the same volume.

O The file is overwritten. In contrast to "D + N", the disk storage previously occupied is not released. It is first overwritten and then extended, if necessary, provided that this is possible. To do this, the existing file must be of the same type as the file to be created, otherwise the same procedure is adopted as for "D + N".

E The file is extended.

--- No file transfer takes place (the transfer request is rejected).

For a better understanding of the processes involved in member transfer, refer to section “Transferring a PO or PDSE data set” on page 81.

Some file attributes (e.g. the file name) are derived from specifications in the NCOPY command, from openFT parameters or from local system defaults.

Most of the attributes of a receive file, however, are determined by the corresponding attributes of the send file.

The following factors determine the extent to which the file attributes of these two files can be mapped onto one another:

– Which attributes are transferred via the link involved?
  A distinction is made between the following types of link:

  Homogeneous link: Link between two openFT for z/OS systems

  Heterogeneous link: Link between openFT for z/OS system and another FT system, e.g. openFT for BS2000

– How well can a receive file attribute be derived from a transferred attribute or a combination of transferred attributes?

  The transferred attributes must be converted into attributes of the generated receive file. The rules governing this procedure are described below in detail.
  The transferred attributes themselves are derived from the attributes of the send file; in many cases, the rules applied are complementary to those described here.

The following send file attributes are transferred via both homogeneous and heterogeneous links:

– file size
– (maximum) record length
– record format (U,V,F)

The following additional attributes can be transferred via a homogeneous link:

– block length
– number of directory blocks
– spanned records attribute
– blocked records attribute
– control character attributes

Attributes of new receive files of any type

The following rules govern the definition of the attributes for a new receive file, regardless of the type of the receive file (PS data set, VSAM file etc. (i.e. cases "N", "D + N" and "O" in the above table):

File structure:

The file structure of the receive file is derived from the prefix specified as part of the file name in the NCOPY command. The following prefixes can be used (see also section “z/OS file names” on page 57):

:S:	for PS
:V:	for VSAM (of the type "entry sequenced")
:L:	for PO or PDSE
:O:	for PO
:E:	for PDSE

In the case of prefixes :L:, :O: and :E: the structure of the file name is also important: :L:filename, :O:filename and :E:filename represent a complete PO or PDSE data set, while :L:filename(membername), :O:filename(membername) and :E:filename(membername) represent a member of a data set.

If complete PO or PDSE data sets are to be transferred homogeneously and in one of the systems openFT < V10 is used then the receive file must always be specified with the corresponding type :O: or :E: in the request since otherwise a PS file is created.

If the file structure of the receive file is not clearly defined by the file name specified in the NCOPY command (i.e. it does not contain a prefix or it contains the prefix :L:, which can stand for PO or PDSE), it is determined by predefined values entered by the FT administrator (for more information, refer to the System Administrator Guide "openFT for z/OS - Installation and Administration") or openFT-internal default values or system defaults. These are:

filename — represents a sequential file; the FT administrator can specify the file structure PS or VSAM (of the type "entry sequenced"); if the FT administrator has not entered a predefined value, the following applies:

- in the event of a homogeneous link between two openFT systems as of V10, the file organization of the send file is taken if it is not a PO or PDSE or an openEdition file.
- in all other cases, PS is the default. (Exception: If the existing file is of the type "VSAM entry sequenced", a VSAM file is again created by default.)

:L:filename
: represents a complete PO or PDSE data set; the FT administrator can specify the file structure PO or PDSE; if the FT administrator has not entered a predefined value, the following default values are used:
  – if the IBM software product DFSMS (Data Facility System-Managed Storage) is not installed, PO is the default
  – if DFSMS is active: the default is specified with an ACS routine or SYS1.PARMLIB (please consult your z/OS system administrator)

filename(membername) or :L:filename(membername)
: represents a member of a PO or PDSE data set; for the selection of the exact file structure, the same principle applies as for a complete PO or PDSE data set (specification :L:filename)

File name:
: The file name is defined by the specification in the NCOPY command (FILE-NAME operand). If the file name specified in the NCOPY command is enclosed in single quotes, it is interpreted as a fully qualified file name (including "first level qualifier"). If the file name specified in the NCOPY command is not enclosed in single quotes, the user ID from the TRANSFER-ADMISSION of the system involved is prefixed to it as the "first level qualifier".

  In the case of a VSAM file, the name formed in this way is used as a cluster name. The data name also required is assigned by IDCAMS (usually the cluster name together with the suffix .DATA). The "first level qualifier" must reference an existing VSAM or ICF catalog as an alias.

Volume:
: A distinction is made here between new files ("N" in the above table) on the one hand, and files which are overwritten ("O") on the other hand.

  In the case of new receive files, the FT administrator can specify the volume (which remains the same for all transfer requests). If the FT administrator has not specified a volume, the volume is determined by the system defaults for new files.

  If, however, the receive file overwrites a file of the same name (not as a result of deletion and re-creation), the volume specified for the old file applies.

Storage allocation:
: For both homogeneous and heterogeneous links, the allocation of disk storage to the receive file is determined by the transferred file size of the send file (see “Primary and secondary allocation” on page 63).

Exception

In cases where an existing PS data set is actually overwritten (entry "O" in the above table), no new disk storage is initially allocated. The storage space already used is overwritten and additional storage, if required, is requested during transfer,

However, the z/OS data management system only permits 16 “extents” for PS and PO files. In the case of PDSE and VSAM files this number is 123. If this is insufficient, file transfer is canceled with an error message. The Data Management System also issues an error message in the job logging facility.

Data access control:

File protection attributes of the send file are not transferred to the receive system.

openFT does not assign any data access control attributes to new files. In particular, no file password is passed on to the system and the so-called "RACF bit" in the DSCB (data set control block) is not set by openFT. If a new file is to be immediately protected against unauthorized access in z/OS, use of the RACF function "generic profile" is recommended (see “Access protection for send and receive files” on page 105).

openFT does not set any retention period for a new file.

Attributes of new receive files of the following types:

– PS data set
– VSAM file
– attributes of new PO data sets created for a receive file of the type PO/PDSE member

The following rules governing the definition of the attributes for a new receive file (i.e. cases "N", "D + N" and "O" in the above table) only apply if the receive file is a PS data set or a VSAM file or if a new PO/PDSE data set was created for a receive file of the type "PO/PDSE member".

(If an entire PO/PDSE data set is being transferred, these rules only determine the attributes of the temporary PS data set created for the transfer. Information on determining the attributes for a PO/PDSE data set created as a receive file is given later.)

Record format:

For both homogeneous and heterogeneous links, the record format attribute of the send file - F(ixed), V(ariable) or U(ndefined) - is transferred.

The following attributes are derived from this for the different types of receive file:

PS data set — The receive file is assigned the transferred record format attribute.

VSAM file

F : AVGLRECL and MAXLRECL both have the same value (AVGLRECL: average logical record length; MAXLRECL: maximum logical record length)

V : the transferred record length determines the MAXLRECL; AVGLRECL = MAXLRECL - 1

U : cannot be mapped → FTR2096

PO/PDSE member

If the entire PO/PDSE data set is new, it is assigned the transferred record format attribute. If the PO/PDSE data set already exists then the attributes of the receive file and the data set must correspond.

openFT for z/OS does not support the record format FS.

Notes

- When transferring a file from a Windows or Unix system to z/OS, the record format attribute transferred depends on the specification for DATA-TYPE in the NCOPY command:

  DATA-TYPE=*CHAR → record format attribute VB
  (default value)

  DATA-TYPE=*BIN → record format attribute U
  (see also the section “Unix and Windows files” on page 90)

- In the following cases, the records of PS data sets and of members of PO or PDSE data sets with undefined record length (RECFORM=U) are split up into smaller records:

  – The length of a record in the send file exceeds the block length. (BLKSIZE) of the receive file to be extended (WRITE-MODE=*EXTEND-FILE).
  – The length of a record in the send file exceeds the maximum length of a unit transferable between send system and receive system.
  – The length of a record in the send file exceeds the maximum block length in the receive system.
  – No record structure is used in the receive system.

  The byte sequence of the data transferred is retained in all these cases.

Block length:

The block length of the send file is only transferred in the case of a homogeneous link.

In the case of a heterogeneous link (and a homogeneous link where the block length of the send file cannot be used for the receive file because of the disk type), a standard block length of 2048 (or a integer multiple of this length) is assumed. (The smallest integer multiple of 2048 into which a record of maximum length - MAXLRECL, see below - will fit is selected as the standard block length.)

The following applies to the different types of receive file:

PS data set — The receive file is created with the above block length.

VSAM file — The above block length is mapped onto the control interval size (CISIZE) as follows:
  – block length < 8192: the value for CISIZE is rounded up to the nearest integer multiple of 512.
  – block length >= 8192: the value for CISIZE is rounded up to the nearest integer multiple of 2048, with 32760 as the maximum.

PO/PDSE member — If an entire new PO/PDSE data set is created, it is assigned the block length described above.

Record length:

In the case of both homogeneous and heterogeneous links, the record length of the send file is transferred. The record length specification is interpreted as follows for the different record formats of the receive file:

F(ixed) Every record has this length.

V(ariable) Record length including length field of 4 bytes

> 0 : The length of a record may not exceed this length.

= 0 : A record can have any length, with 32752 byte as the maximum (without record length field). However, there are restrictions depending on the file type, see below.

U(ndefined) No meaning (= 0)

A maximum length (≠ 0) **must** be specified in z/OS, however, for format V in the case of PS, PO and VSAM files. For this reason, if the record format = V and the record length transferred = 0 (as may be the case for heterogeneous links), openFT for z/OS must specify a value it can use for the maximum record length. openFT for z/OS selects this value in accordance with the type of receive file. The value 259 (255 + 4 bytes record length field) is used as the default value.

The following applies to the different types of receive files:

PS data set:

record length transferred > 0:
The file is created with this (maximum) record length.

record length transferred = 0:

U format : OK.

F format : This combination of attributes may not occur in the send file.

V format: If the longest record in the file is not longer than 259 bytes (incl. 4-byte record length field), then the file is created with a record length of 259 bytes. If the longest record length in the file (including the 4-byte record length field) is greater than 259 and does not exceed the block length - 4, this is taken as the maximum record length. Records longer than this value cannot therefore be transferred and result in a cancellation of the file transfer operation, accompanied by an error message.

If files are received from BS2000, z/OS uses the BS2000 block size as the record length.

PO/PDSE member

The following applies if the entire PO/PDSE data set is newly created:

record length transferred > 0:
The PO/PDSE data set is created with this (maximum) record length.

record length transferred = 0:

U format : OK.

F format : This combination of attributes may not occur in the send file.

V format: The PO/PDSE data set is created with a maximum record length of 259 (255 + 4).

VSAM file:

record length transferred > 0:

F format: MAXLRECL = record length transferred
AVGLRECL = record length transferred

V format: MAXRECL = record length transferred - 4
(since the length field is not counted)
AVGLRECL = MAXLRECL - 1

record length transferred = 0:

F format: This combination of attributes may not occur in the send file.

V format: MAXLRECL = 32752 (the length field is not counted)
AVGLRECL = 255 (since the length field is not counted)

If, during transfer, the length of a record received exceeds the maximum record length created for the receive file, the transfer is canceled with an error message.

Notes

- The maximum length of the records that are to be transferred may not exceed the following values:
  – 32760 byte for files with fixed record lengths
  – 32752 byte for files with variable record lengths (record length without record length field)
  – for openFT ≤ V10.0: 32248 byte with compromised transfer (COMPRESS = *BYTE-REPETITION)
- So it can be assumed that, send files from remote systems which are to be transferred in format V with a record length of = 0 cannot be stored in newly created receive files under the following circumstances.

  – The receive file is a newly created PS data set, and the send file contains records with more than 2040 characters (net) in the case of BS2000 send files, or 255 characters in the case of Windows or Unix send files.
  – The receive file is a PO member in a newly created PO/PDSE data set, and the send file contains records with more than 255 characters (net).

  In either case, the transfer is canceled with an error message.
- In z/OS, the same maximum record length applies to **all** members of a PO/PDSE data set. For LMS libraries under BS2000, however, the maximum record length is member-specific. This means that the following situation may occur:

  An LMS member is transferred to a PO/PDSE member. The PO/PDSE data set does not yet exist. It is therefore created with the maximum record length transferred.

  A second member of the same LMS library is now to be transferred to the same PO/PDSE data set. If this LMS member (whose maximum record length may differ from that of the first member) contains at least one record whose length exceeds the maximum record length of the first LMS member, the transfer is canceled with an error message. This happens because the length of this record is not compatible with the maximum record length of the PO/PDSE data set, which was determined by the maximum record length of the first member to be transferred.

  If, therefore, a number of members from the same LMS library are to be transferred to members of the same PO/PDSE data set, the member with the greatest maximum record length should be transferred first.

Spanned records attribute:

The spanned records attribute of the send file is only transferred in the case of homogeneous links. The spanned records attribute transferred is only taken into account when creating non-VSAM files (PS data set, PO/PDSE member) with variable record length. It is then either set or not set in the receive file as in the send file.

In the case of heterogeneous file transfer, the spanned records attribute is only set if LRECL > BLKSIZE-4.

If the partner system specifies the value 0 as the record length of the send file, the spanned records attribute is never set.

If a record with LRECL > BLKSIZE-4 is to be written to a file (PS file or PO/PDSE library) with variable record length for which the spanned records attribute is **not** set, the transfer is canceled with an error message.

In the case of VSAM files, the spanned records attribute corresponds to the file attribute SPANNED. It is only set for a VSAM receive file if the record length transferred (including the length byte) is greater than the value already rounded up for CISIZE - 7 (VSAM block overhead).

If, however, during homogeneous transfer, a non-VSAM file is created as a receive file and the send file is a VSAM file with the SPANNED attribute, the spanned records attribute is also set for the receive file.

Blocked records attribute:

The blocked records attribute can only be set for non-VSAM files (PS data set, PO/PDSE member). The blocked records attribute of the send file is only transferred in the case of a homogeneous link. It is then either set or not set for the receive file, as for the send file.

In the case of heterogeneous file transfer, the blocked records attribute is always set.

Note

If, during homogeneous transfer, a non-VSAM file is created as a receive file and the send file is a VSAM file, the blocked records attribute is always set for the receive file (in this case, VSAM is interpreted as "blocked").

Control character attributes:

Control character attributes (**A**NSI and **M**achine control characters) are only taken into account in non-VSAM files.

The control character attributes of the send file are transferred and accepted when the receive file is created only in the case of a homogeneous link.

Number of directory blocks:

PS data set, VSAM file: Not applicable

PO/PDSE member

If an entire new PO/PDSE data set is created, the value specified by the FT administrator for the number of directory blocks is used. If the FT administrator does not specify a value for the number of directory blocks, the default value 20 is used.

One important attribute of a PO data set, however, is not contained in the "unloaded data". This attribute must be defined in a different way:

Number of directory blocks:

If the attribute "number of directory blocks" transferred via a homogeneous link contains a value other than zero, the new PO data set is created with the corresponding number of directory blocks. This applies if the send file also contains an entire PO data set identified by the prefix ":L:" in the NCOPY command). In this case, the receive file is assigned the same number of directory blocks as the send file.

If, however, the attribute "number of directory blocks" transferred does not contain a valid value (e.g. because the send file is a PS data set containing a PO data set in "unloaded" format), the number of directory blocks for the new PO data set is defined by the value specified by the FT administrator or the default value 20, as for the transfer of a PO member.

3.3.2.4 Transferring a PO or PDSE member

The send or receive file for transfer with openFT may be an individual member of a PO or a PDSE data set. In this case, the same file contents are transferred as when a sequential file is transferred. In contrast to the transfer of an entire PO data set (see below), no directory information is transferred. In this case, the following specifications must be made for the **file name** in the NCOPY command:

fully qualified: 'data-set-name(member-name)'
partially qualified: data-set-name(member-name)

The data set name can contain one of the prefixes ":L:", ":O:" or ":E:". If it does not have a prefix or has the prefix ":L:", the name represents any data set (PO or PDSE). To access a specific PO member, you must use the prefix ":O:"; for a PDSE member, you must use the prefix ":E:".

You may specify *REPLACE-FILE, *NEW-FILE and *EXTEND-FILE for the **WRITE-MODE** operand. In this case, you are referring to the individual member. An existing PO or PDSE data set is always retained. If necessary, the PO or PDSE data set itself is also created.

You can specify the type of the data set (PO or PDSE) using the prefixes mentioned above (":O:" for PO, ":E:" for PDSE). If you do not specify a prefix, or you specify the prefix ":L:", a default value defined by the FT administrator is taken as the type for the newly created data set.

These specifications have the following meaning:

*REPLACE-FILE:	The member is written to the data set even if a member with the same name already exists. For a PO data set applies: The new member is then located at the end of the PO data set; there may be a gap in place of the old member.
*NEW-FILE:	Transfer takes place only if no member with the same name exists.
*EXTEND-FILE:	The contents of any existing member with the same name are copied. The transferred data is appended and the resulting member is written to the data set. For a PO data set applies: The extended member is then located at the end of the PO data set; there may be a gap in place of the old member.

It is clear from this description that the transfer of individual members may result in gaps in the PO data set.

If a new PO or PDSE data set is created on receipt of a member, then the primary allocation corresponds approximately to twice the size of the send file and the secondary allocation to four times its size.

If members are written to an existing data set then the primary and secondary allocations remain unchanged.

If necessary, you must make sure that the allocation for a PO data set that is to be received is sufficiently large. If a large number of members are to be entered then sufficient directory blocks must be available. If members are frequently replaced or extended then a PO data set should be periodically compressed. No compression is required for PDSE data sets.

File transfer requests with read access to a member can be restarted an unlimited number of times. Requests that write to a member can also be restarted. However, when transferring PDSE data set members during restarted requests, openFT restarts at the beginning of the file. Requests that extend existing members are only permitted for PO data set members. Please also note that if a restart has to be performed during transfer of a PO data set member, this member is always appended to the PO data set. It is therefore advisable to compress this type of PO data set from time to time.

In the event that, on receipt of a PO or PDSE member, the data set must be created again, and this creation attempt fails, the file transfer will be aborted and a error message output.

Possible reasons for this failure include, among other things, disk bottleneck or a faulty specification for one of the SMS classes. If this happens, you should speak to your FT administrator.

3.3.2.5 Transferring a PO or PDSE data set

openFT can transfer an entire PO or PDSE data set to another z/OS computer in a single transfer request. This is only possible, however, if openFT is "APF authorized"; please consult your FT administrator regarding this authorization. In the absence of APF authorization, any attempt to transfer a complete PO or PDSE data set is rejected.

If you want to transfer an entire PO or PDSE data set and an openFT version < V10 is running on one of the partners, you must prefix the **file name** with one of the character strings ":L:", ":O:" or ":E:" in the NCOPY command as follows:

- for any data set (PO or PDSE):

  fully qualified file name: ':L:data-set-name'
  partially qualified file name: :L:data-set-name

- for access to a specific PO data set:

  fully qualified file name: ':O:data-set-name'
  partially qualified file name: :O:data-set-name

- for access to a specific PDSE data set:

  fully qualified file name: ':E:data-set-name'
  partially qualified file name: :E:data-set-name

When transferring entire PO or PDSE data sets, you can only specify *NEW-FILE or *REPLACE-FILE as the **WRITE-MODE**. These specifications have the following meaning:

*REPLACE-FILE:	If a data set with the same name already exists, it is overwritten in its entirety.
*NEW-FILE:	Transfer takes place only if no data set of the same name already exists.

The transfer of complete PO or PDSE data sets with WRITE-MODE=*EXTEND-FILE is aborted with an following message.

openFT uses the IBM utility IEBCOPY for transferring entire PO or PDSE data sets. File transfer takes place as follows:

– The send file identified as a PO or PDSE data set is transferred to a temporary PS data set by means of IEBCOPY-Unload.

– This temporary PS data set is transferred to the remote system.

– The receive file identified as a PO or PDSE data set is generated from this temporary PS data set by means of IEBCOPY-Load in the remote system. You can specify the type of the data set (PO or PDSE) using the prefixes mentioned above (":O:" for PO, ":E:" for PDSE). If you do not specify a prefix, or you specify the prefix ":L:", the dataset type of

the send file (PO or PDSE) is used as the type of the newly created dataset if openFT as of V10 is running on the z/OS partner system, otherwise the default value specified by the FT administrator (prefix ":L:") is used.

– The temporary PS data sets are then deleted from both systems.

This procedure ensures that, when transferring entire PO or PDSE data sets, all the structure information is also transferred. This information is also retained when copying a PO or PDSE data set using IEBCOPY. Files containing load modules can also be transferred in this way.

When a PO or PDSE data set is transferred using IEBCOPY, a temporary PS data set is created in the send and receive system as a buffer for the file in "unloaded" format. This PS data set has the following attributes:

File name: `'transuid.podsname.U'` where

transuid	is the user ID which was specified in the TRANSFER-ADMISSION for the system involved. This also applies if the name of the PO or PDSE data set was specified in its fully qualified form in the NCOPY command. In this case, the "first level qualifier" for the temporary PS data set is replaced by this user ID.
pdsname	Partially qualified name of the PO or PDSE data set.
.U	This suffix identifies the temporary PS data set. For this reason, the length of the PO or PDSE data set itself may not exceed 42 characters (fully qualified) or 34 characters (partially qualified).
Volume:	openFT creates the temporary PS files on a volume specified by the FT administrator (the same volume is used for all transfer requests). If the FT administrator has not specified a volume, the default volume defined for new files on the system involved applies.
Storage requirements:	About the same as for the associated PO or PDSE data set
Other attributes:	Default values generated by IEBCOPY, i.e.: RECFORM= VS, BLKSIZE= max(284, PO-DS-BLKSZIZE + PO-KEYLEN + 20), RECSIZE= BLKSIZE - 4

Please refer to the note on data protection at the end of this section on page 84.

Notes on transferring entire PO or PDSE data sets

- No file may exist with the same name as the temporary PS data set required ('transuid.podsname.U') in either of the two systems involved. Otherwise, the transfer request is rejected or canceled with an appropriate error message.

  A file with the same name as the temporary PS data set required ('transuid.podsname.U') may exist in the following case:

  An NCOPY command is issued under two different user IDs. The user ID of the TRANSFER-ADMISSION (transuid) and the **partially** qualified name (podsname) of the PO data set (as send or receive file) are identical in both commands. This is also the case if the name of the PO or PDSE data set is specified in its fully qualified form in each case, and only the "first level qualifier" is different (see above). The temporary PS data sets required then have the same name in both transfer requests. In this case, you should wait until the first transfer request is complete and then repeat your transfer request.

- The user ID (transuid) specified in the TRANSFER-ADMISSION must be authorized to create a file with the name of the temporary PS data set required ('transuid.podsname.U'). This may be rejected by RACF, for example, with the usual error messages.

- If the attempt to swap out the PO or PDSE dataset to a PS dataset using IEBCOPY-Unload fails on the send side, the transfer request is canceled with an error message.

- If necessary, you must make sure that the allocation for a PO data set that is to be received is sufficiently large. If a large number of members are to be entered then sufficient directory blocks must be available. If members are frequently replaced or extended then a PO data set should be periodically compressed.

- If there is not enough storage space in the receive system for the PS data set transferred, the transfer request is canceled and an error message is issued.

- If the attempt to create the PO or PDSE dataset from the transferred PS dataset using IEBCOPY-Load fails on the receive side, the transfer request is canceled with an error message.

  In this case, the transfer itself is already complete.

Note on data protection

The temporary PS files (files with the suffix ".U", see above) created by openFT during the transfer of PO or PDSE data sets may contain confidential information. For this reason, you may want the contents of these files to be physically deleted when the files are deleted by openFT.

Any openFT user can control this, e.g. by using the IBM product RACF to create a "generic profile" for all files created internally by openFT for this user which are to be physically deleted after use. The parameter "Erase when deleted" must then be set to "YES" for this profile.

WARNING!
Since this procedure causes files to be overwritten with dummy information in z/OS systems, it involves a high degree of CPU and I/O time consumption.

Example

The user with the user ID MILLER, who also uses this ID in the TRANSFER-ADMISSION of the NCOPY command, wants the intermediate files, created internally by openFT for the transfer of entire PO or PDSE data sets, to be physically deleted. To this end, the user creates the following "generic profile" (RACF):

'MILLER.*.U'

The parameter "Erase when deleted" is set to "YES" for this profile.

3.3.2.6 Transferring a generation data set

openFT transfers absolute and relative generation data sets from PS data sets. openFT does not, however, create new generation data groups; these must already exist before transfer takes place. In addition, a DSCB model (DSCB: Data Set Control Block) must exist on the same volume as contains the generation data group. (A DSCB model is created automatically and correctly if a generation data group is created in a user catalog with a TSO command. This is not the case if the generation data group is created in the "master catalog". In this event, a DSCB model must be explicitly created with JES JCL, as described in the IBM documentation; see VSAM Catalog Administration, Access Method Reference, for example.)

openFT cannot transfer entire generation data groups.

Absolute generation data set

The name of an **absolute** generation data set must be specified in the NCOPY command in the usual format for TSO:

fully qualified file name: `'generationgroup.GmmmmVnn'`
partially qualified file name: `generationgroup.GmmmmVnn`

generationgroup: is the name of the generation data group
max. 35 characters with "first level qualifier" (fully qualified)
max. 27 characters without "first level qualifier" (partially qualified)

mmmm is the absolute generation number (between 0000 and 9999)

nn is the version number (between 00 and 99)

Relative generation data set

The name of a relative generation data set must also be specified in the usual format for TSO:

fully qualified file name: `'generationgroup(0)'` or `'generationgroup(±m)'`
partially qualified file name: `generationgroup(0)` or `generationgroup(±n)`

generation group: is the name of the generation data group (see above)

(...) is the relative generation number:

(0) current generation

(±m) m = 1..255 (fully qualified specification)

(±n) n = 1..99 (partially qualified specification)

In the following description, "(±n)" is used to represent both "(±m)" and "(±n)".

If a relative generation data set name is specified, openFT forms the absolute generation data set name according to the following rules:

Generation number

The absolute generation number is formed from the current generation number and the relative generation number specified:

relative generation number (0):

absolute generation number	=	current generation number

relative generation number (-n):

absolute generation number	=	generation number of the "n-th predecessor" of the current generation data set (determined using the LISTCAT command)

relative generation number (+n):

absolute generation number	=	current generation number + n modulo 9999

(The current generation number for an empty generation group is G0000.)

For files in the local system, the relative generation number is converted to an absolute generation number when the request is accepted. For files in the remote system, this conversion takes place at the start of request processing. In this context, please also refer to the notes at the end of this section.

Version number

When a receive file is first created, the version number is set to V00.

If the receive file already exists, the previous file name is used for WRITE-MODE=*REPLACE or*EXTEND (no increment of the version number).

Existence of send and receive files when transferring relative generation data sets

- If a relative generation data set is specified as the **send file**, then it must, of course, already exist.

  The specification of a non-existent file generation as a send file (e.g. a positive relative file generation(+n)) therefore results in an error message.

- New generation data sets (**receive file**) are created only if a relative generation data set > 0 is specified, regardless of the WRITE-MODE. In particular, it is not possible to create a "current" generation data set (0) in an empty generation data group.

  If you specify a relative file generation≤ 0 for a receive file while also specifying WRITE-MODE=*NEW, an error message is output.

  You may specify a relative generation data set ≤ 0 for a receive file if WRITE-MODE=*REPLACE or *EXTEND is also specified; only an **existing** generation data set, however, can be replaced or extended. If the corresponding generation data set does not exist, an error message is issued.

Causes of the error message FTR0020

The message FTR0020 is issued in many of the error situations mentioned above that can arise in the context of file generation:

```
FTR0020 OPENFT: 'file' not found.
```

This error message may be caused by:

– The generation data group of the send or receive file does not exist.

– The relative generation number (0) was specified but the generation group of the send or receive file is empty.

– A relative generation number (-n) was specified but the corresponding generation data set (send or receive file) does not exist.

Please also observe the following **notes** on transferring relative generation data sets:

- If the absolute generation number G9999 is exceeded, modulo 9999 applies (see above), i.e. the generation number reverts to zero in the sequence of counting. This can have unexpected results if an absolute generation number calculated in this way has already been assigned to existing generations.
- When a new generation is created, the oldest generation or all previous generations may be deleted (depending on the LIMIT, EMPTY/NOEMPTY and SCRATCH/NOSCRATCH parameters specified in the DEFINE-GENERATIONDATAGROUP command when the generation data group was created).
- For files in the local system, the relative generation data set name is converted to an absolute generation data set name when the request is accepted. For files in the remote system, this conversion takes place at the start of request processing. Once a generation name has been assigned, it remains the same throughout request processing. This has the following results:
  – If, once the absolute generation data set name has been formed, new file generations are created (e.g. by a batch job executing in parallel), the resulting displacement of the "current" generation data set number is not taken into account.

  - The relative generation data set name appears in the result list and in the transfer logging function (FT administrator function, see the "System Administrator Guide openFT for z/OS and MVS - Installation and Administration") exactly as it was specified in the NCOPY command.
  - Similarly, in a NSTATUS or NCANCEL command, you must specify the relative generation data set name exactly as it was specified in the original NCOPY command.
- There is no way of ensuring that two transfer requests cannot access the same generation data group. If this occurs, the generation numbering may be displaced unexpectedly.
- In an FT system which does not support the specification of relative generation data set names for REM=*MSP, the REM=*ANY syntax can be used to access a relative generation data set as a send or receive file in a remote system of type openFT for z/OS.
- Inbound file management requests cannot access relative file generations.

3.3.3 Unix and Windows files

Files in Unix systems and Windows systems, like POSIX files in BS2000/OSD, have no structure and no file attributes that provide information on the coding. Although they have no structure either, Windows files can be distinguished on the basis of their file extensions (e.g. “txt” for text and “exe” for executable files).

For transfer with Windows or Unix systems, you can therefore define the following file types:

– text
– unstructured binary data
– binary data structured in records (user format)

Text format

A file that is sent in text format from Windows or Unix systems, must be a pure text file with a record structure defined by linefeed characters in Unix systems or Carriage Return and linefeed in Windows. The length of a line is limited, e.g. 98403 bytes in Windows systems. The end-of-line character is removed from every line.

During transfers from BS2000/OSD or z/OS to Windows or Unix systems, the end-of-line character is inserted into the sentence length already in the remote system. The text and the sentence lengths are preserved. The line length is restricted, e.g. to 98304 bytes in Windows systems. The maximum sentence length during a text file transfer depends on the operating system.

When communicating with partner systems as of openFT V10, it is also possible to transfer Unicode files; see section “Transferring 7-bit, 8-bit and Unicode files” on page 96.

Tabulator and blank line expansion

During transfers of text files, openFT carries out a tabulator and blank line expansion if necessary. This means that blank characters will be transferred instead of a tabulator, and a line with a blank character will be transferred instead of a blank line. During this, the following cases will be different for openFT partners:

Initiator	Direction	Responder	Expansion (yes/no)
Unix system, Windows system	Send	Unix system, Windows system	no, optional yes[1]
Unix system, Windows system	Fetch	Unix system, Windows system	no
Unix system, Windows system	Send	BS2000, z/OS	yes, optional no[1]
Unix system, Windows system	Fetch	BS2000, z/OS	no (not relevant)
BS2000, z/OS	Send	Unix system, Windows system	no (not relevant)
BS2000, z/OS	Fetch	Unix system, Windows system	yes (at the initiator)
BS2000, z/OS	Send and Fetch	BS2000, z/OS	no

[1] The expansion can be explicitly enabled or disabled in Unix systems and Windows system during the request.

Binary transfer with openFT for Unix and Windows systems

openFT for Unix systems and openFT for Windows differentiate two file formats for binary transfer (for more information, see section “Unix and Windows files” on page 89):

– Binary format: The file contents are considered to be an unstructured sequence of binary data.

– User format: The file contents are considered to be a sequence of records containing binary data. In this case, a Unix or Windows file is divided into records by means of record length fields.

This section describes the transfer of these files between openFT for z/OS and openFT for Unix systems. The process for openFT for Windows is similar.

If the transfer is **initiated in the Unix system**, the *filetype* parameter can be used in the ft or ncopy command to specify how openFT for Unix is to read binary data from a send file or write binary data to a receive file:

filetype = b: binary format

filetype = u: user format

If the transfer is **initiated in the** openFT **system for z/OS**, the procedure is different depending on whether a binary file is being sent (TRANSFER-DIRECTION=*TO-PARTNER) or fetched (TRANSFER-DIRECTION=*FROM-PARTNER).

Sending a binary file to the Unix system

If the transfer is initiated in the openFT system for z/OS and a binary file is to be sent to the Unix system, only DATA-TYPE=*BINARY or DATA-TYPE=*USER can be specified. The Unix receive file will have the following file format, depending on the record format of the z/OS send file:

Record format of z/OS send file	**File format of Unix receive file**
F(B)	Binary format (no record length fields)
V(B)	User format (with record length fields)
U	Binary format (no record length fields)

Fetching a binary file from the openFT for Unix system

If the file transfer is initiated in the openFT for z/OS system and a file with **DATA-TYPE=*BINARY** is fetched from the Unix system, the z/OS receive file is recreated with record format U if it does not yet exist, or an existing receive file is overwritten. The contents of the Unix send file are then transferred as an unstructured sequence of binary data and are stored in the z/OS receive file. If a Unix send file is in user format, the record structure is lost.

If the record structure of a Unix send file in user format is to remain unchanged when a file transfer is initiated in the openFT system for z/OS, the following options are available:

a) If the receive file already exists in the z/OS system, and it has the record format V(B) and is being extended to include the contents of a Unix send file in user format (WRITE-MODE=*EXTEND-FILE), the record structure of the send file is unchanged when the transfer is carried out with **DATA-TYPE=*BINARY** or **DATA-TYPE=*USER**. You can take advantage of this by creating an empty receive file in the z/OS system with the record format V(B) before file transfer; you must then specify DATA-TYPE=*BINARY and WRITE-MODE=*EXTEND-FILE in the NCOPY command.

b) If, when fetching a Unix send file in user format, a PS data set is to be created with record format = V(B) or an existing data set is to be overwritten with record format V(B), so that the record structure of the Unix send file is unchanged, **DATA-TYPE=*USER** must be specified in the NCOPY command. The following conditions must then be met:

   – The Unix send file must be in user format; otherwise, the result of the file transfer request cannot be predicted. In this case (and also when *-u* is specified in a to Unix system addressed file transfer command, i.e. *ft* or *ncopy*), openFT for Unix interprets any two bytes of the data flow contained in the send file, which can contain any information, as "record length fields".

– In the z/OS system it is only possible to specify one PS file or one PO/PDSE member. If the library already exists, it must have the record format V(B). The maximum record length of the receive file is adapter to that of the send file (see also page 72ff). If the send file does not possess a maximum record length not equal to 0 as an attribute and no maximum record length is specified in the file transfer request then the following applies:

  When a PO/PDSE library is created or a PS file is overwritten, it is possible to write records of length up to 259 bytes (including the 4-byte record length field).

  When a PO/PDSE member is written to an existing library, or when a PS file is extended, the specification for the receive file is definitive. However, there is an upper limit of 2044 bytes (including the record length field).

3.3.4 Transfer of various file types

Besides complete transfer of the contents of a file, file transfer also aims at producing an authentic representation of the file structure. If identical structures are mapped to each other, as is the case with homogeneous links, authenticity is achieved without any problem, i.e. the binary code and the character representation are identical in the send and receive system. With heterogeneous links, however, it is usually not possible to obtain the binary code and the character representation in the receive system unchanged. For this reason, a distinction is made between text and binary transfer for file transfer with openFT.

Text transfer

Text transfer is character-oriented, i.e. the presentation of the characters is retained. This applies both to characters in single-byte code such as ISO 8859 and to Unicode characters which are represented by multiple bytes. The record structure of the text file is matched to the system conventions of the receive system when the file reaches the receive system.

The “useful data” of a file to be sent per text transfer must not contain any characters which the receive system could interpret as control characters, e.g. X‘15‘ (EBCDIC linefeed) and X‘0A‘ (ASCII linefeed).

In the case of text transfers, openFT for z/OS can use file-specific conversion tables that are selected via the file name; please consult your FT administrator.

Binary transfer

Binary transfer is carried out such that the coding (binary representation) of the characters is retained. The design of the record structure can be controlled. In this way, openFT matches the record structure with the record structure of the receive system (system-conformant record structure). With the original record structure, the structure of the send system is retained. Furthermore, it is possible to employ your own system-dependent record structures using the FT-specific user format.

It is not possible to fetch binary format files with fixed length or variable length records using the FTP protocol. In particular, this also applies to the output of file transfers with preprocessing on BS2000 or z/OS and the output from commands executed using *ftexec* on BS2000 or z/OS. In this case, you must either transfer files in text format or use a different transfer protocol (openFT).

ISAM and PAM files can be transferred between BS2000 systems and other systems as follows:

– in transparent format, see page 94

– by specifying the target format, see the section “Heterogeneous transfer of PAM and ISAM files” on page 94

Record by record transfer

openFT for z/OS usually takes into account the record structure during file transfer (exception: transfer of an unstructured sequence of binary data during file transfer with Windows and Unix systems). With record-by-record transfer the maximum length of the records to be transferred may not exceed the following values:

– 32760 byte in files with fixed-length records
– 32752 byte for files with variable record lengths (record length without record length field)
– 32248 byte for compressed transfer (COMPRESS = *BYTE)

Transfer with transparent file format

A special case is the transparent file format. This file format provides you with the option of passing through any BS2000 files over a variety of FT platforms to a BS2000 system, while retaining their original file attributes. This procedure is useful for distributing BS2000 files from a Unix based server or Windows server to BS2000 systems, for example. From the point of view of the intermediate processor, the files received, which cannot be used by this processor, are binary files. These files are then set up on the receive processor with their original attributes by openFT for BS2000/OSD.

openFT for z/OS can act as a buffer for BS2000 files in transparent file format. However the transfer of these files must be initiated in the BS2000.

openFT does not provide any direct means of transferring z/OS files true-to-format (“transparent”) over FT platforms other than z/OS. Hover, you can use the TSO command XMIT to pack files in a neutral format and transfer them in this format as binary files with a fixed record length of 80 bytes. You do this, for example, by specifying ***-r=f80*** in the openFT *ft* command in Windows or on a Unix system. The file can then be unpacked at the target system using the TSO command RECEIVE.

Heterogeneous transfer of PAM and ISAM files

You can transfer BS2000 PAM files onto a foreign system such as a Unix or Windows system or to z/OS and then retrieve them to BS2000 and store them there as PAM files. The foreign system can also have the initiative for this request. You can also transfer ISAM files from a BS2000 systems onto a foreign system. In all cases, the prerequisite for this is that openFT as of V11 is running on the foreign system.

To do this, proceed as follows:

- Transferring a PAM file from BS2000 to a foreign system
  Specify "sequential" as the target format in the transfer request.
- Storing a binary file from a foreign system as a PAM file in BS2000
  Specify "binär" as the file format and "block-structured" as the target format in the transfer request.
- Transferring an ISAM file to the foreign system
  Specify "sequential" as the target format in the transfer request. The ISAM keys are integral parts of the records that are read and are therefore transferred with the file. However, they no longer have any function as index keys. The record format of the target file is to be the same as that of the ISAM file. The format used is compatible with FTP-BS2000.

3.3.5 Migrated files

openFT can access migrated files in BS2000/OSD and z/OS. This means that you can view the properties of such files, and transfer, delete or overwrite them. To do this, openFT as of V10 must be used in the system involved. The following applies to the mainframe systems used:

- In BS2000 systems, the file must be a DMS file. It is not possible to directly transfer individual elements of a migrated library. To do this, the migrated library must first be read in. This can, for instance, be done during preprocessing and postprocessing or using /EXEC-REM-CMD or *ftexec*.
- In z/OS systems, z/OS as of V1.7 must be used, because the necessary values are only returned at the system interface as of this version.

3.4 Transferring 7-bit, 8-bit and Unicode files

In computers with different operating systems, the individual characters, letters and digits are represented internally ("coded") in different ways. In addition, it is possible to use different character sets in these various systems. The content of a text file is interpreted differently depending on the character set used and is output accordingly on the screen or at the printer.

openFT makes it possible to assign various single-byte character sets (7-bit and 8-bit) as well as multi-byte character sets (Unicode) to text files.

3.4.1 Code tables and coded character sets (CCS)

The concept of so-called "Coded Character Sets" (CCS) is supported for openFT partners. A CCS defines a character set and the coding of these characters in the file. A CCS is assigned a name of up to 8 characters in length via which the CCS can be addressed.

In Unix and Windows systems and in z/OS systems, the standard character set is defined via openFT operating parameters. In BS2000/OSD systems, the character set defined in the system settings is used by default (HOSTCODE system variable). However, in BS2000/OSD, it is also possible to assign a file a specific CCS via the catalog entry, see also "openFT for BS2000/OSD - User Guide".

Moreover, for each individual file transfer, you can specify a CCS separately for the local and remote files, see section “Specifying the CCS on a transfer request” on page 97.

Frequently used example CCS's are:

ISO88591
: Character set in accordance with the definition contained in ISO standard 8859-1, ASCII-oriented coding in accordance with ISO standard 8859-1.

EDF041
: Character set in accordance with the definition contained in ISO standard 8859-1, EBCDIC-oriented coding in accordance with Fujitsu definition DF04-1.

IBM1047
: Character set as defined in ISO 8859-1. IBM1047 is an EBCDIC-based encoding compliant with the IBM definition IBM1047 and used as default in z/OS systems.

UTF8
: The character set is Unicode, the UTF-8 multi-byte coding defined in the Unicode standard is used.

UTF16
: The character set is Unicode, the UTF16 16-bit coding defined in the Unicode standard is used.

CP1252

The character set is a Microsoft-defined superset of the character set specified in ISO standard 8859-1. The codings of CP1252 and ISO 8859-1 are identical for the shared characters from the ASCII 7-bit character set. The other characters defined by Microsoft (including the Euro symbol) are present in the code range 0x80-0x9F which is not used by ISO 8859-1.

3.4.2 Specifying the CCS on a transfer request

When transferring text files, you can specify a request-specific CCS for both the local system and the remote system:

– The CODED-CHARACTER-SET operand in the LOCAL-PARAMETER of the transfer command specifies the CCS for reading or writing the local file.

– The CODED-CHARACTER-SET operand in the REMOTE-PARAMETER of the transfer command specifies the CCS for reading or writing the remote file.

If the remote file is a BS2000 file to which a CCS name has already been assigned via the catalog entry then you may not specify a CCS name that is different from this.

The remote CCS name is only supported for the openFT protocol and for partners as of V10.

If the local or remote CCS name is omitted then the default settings for the relevant system apply:

– openFT operating parameters in a Unix system, Windows system or z/OS system,

– in a BS2000 system, the CCS corresponding to the file's catalog entry (if present), otherwise the HOSTCODE system parameter.

In z/OS, a particular CCS can be assigned to files on the basis of a setting in the FT parameter library.

Caution!
If you save the file in a character set which is not a superset of the character set originally used for the file then information is lost! All characters that cannot be mapped to the newly assigned character set are represented by a replacement character. This type of conversion cannot be undone without data loss!

3.4.3 Data conversion

The type of data conversion depends on the openFT version that is used on the partner system.

Data conversion in the case of partners as of V10

Depending on the code class (ISO 8859 or DF04) and code variant n (n=1...10, 13, 15) of the local CCS, openFT as of V10 sends the data encoded in ISO 8859-n, DF04-n or UTF-8.

This has the following effect depending on the partner system:

– In the case of transferring files belonging to the code classes ISO 8859 or DF04 between Unix and Windows systems and BS2000 or z/OS, recoding is performed at the receiving system (if necessary).

– UTF-8 files are recoded at the receiving system (if necessary). Files to which a CCS is assigned that belongs neither to the ISO 8859 code class not to DF04 are recoded into UTF-8 at the sending system and into the CCS of the target file at the receiving system (if necessary).

– UTF-16 files are recoded into UTF-8 at the sending system and into UTF-16 at the receiving system (if this is requested).

– UTF-16 files generated by openFT possess the endian model and line break convention (LF or CRLF) appropriate to the platform in question.

– UTF-8 files generated by openFT possess the line break convention appropriate to the platform in question.

Data conversion in the case of partners < V10

The transferred data is coded in DF04-n. I.e. when file transfer is performed with openFT partners, the data is transferred in EBCDIC format (corresponds to CCS DF04-n). EBCDIC is used, for example, in BS2000/OSD.

3.5 Entries for the remote system

With the entries for the remote system, you define the partner system and inform it of your transfer admission for a login name in the partner system.

openFT recognizes three types of partner:

- Named partners: All partners that are entered with names in the partner list.
- Registered dynamic partners: All partners that are entered without names in the partner list.
- Free dynamic partners: All partners that are not entered in the partner list.

3.5.1 Defining the partner computer

The partner system is the remote system with which files are to be exchanged. By specifying the transfer direction you stipulate whether the partner is to send or to receive files. You address the partner system via a partner name or its partner address ("**dynamic partners**").

The FT administrator may deactivate the use of dynamic partners for security reasons. In this case, you may only use partner names from the partner list.

Partner name

A partner name is a name of 8 characters or less which is assigned by the FT administrator when including a partner system in the partner list. This approach should primarily be used for partner systems which are frequently communicated with.

Partner address

If the FT administrator has not assigned a partner name or if you do not know the name, you can address a partner host using the partner address. A partner address has the following structure:

[protocol://]host[:[port].[tsel].[ssel].[psel]]

host (= computer name, see page 100) is mandatory; all other specifications are optional. In many cases, the other specifications are covered by the default values, so that the host name suffices as the partner address, see “Examples” on page 101. Final ’.’ or ’:’ can be omitted.

The individual components of the address have the following meanings:

protocol://
: Protocol stack via which the partner is addressed. Possible values for *protocol* (uppercase and lowercase are not distinguished):

  openft openFT partner, i.e. communication takes place over the openFT protocol.

  ftp FTP partner, i.e. communication takes place over the FTP protocol.

  ftadm ADM partner, i.e. communication takes place over the FTADM protocol for remote administration and ADM traps.

  Default value: **openft**

host
: Computer name via which the partner is addressed. Possible entries:

  – internet host name (e.g. DNS name), length 1 to 80 characters, up to 24 characters for z/OS partner systems

  – TNS name from the z/OS library (TNSTCPIP member), up to 8 characters in length.

  – SNA LU name, length 1 to 8 characters

  – IPv4 address with the prefix %ip, i.e. for example %ip139.22.33.44
  The IP address must always be specified as a sequence of decimal numbers separated by dots and without leading zeros.

port
: When a connection is established over TCP/IP, you can specify the port name under which the file transfer application can be accessed in the partner system.
  Permitted values: 1 to 65535;

  In the case of an SNA-LU connection, (*host* = LU name) you must specify the value *sna* for the port number.

  Default value: **1100** for openFT partners
  A different default value can also be set in the operating parameters using FTMODOPT.

  21 for FTP partners

  11000 for ADM partners

tsel
: Transport selector under which the file transfer application is available in the partner system. The transport selector is only relevant for openFT and FTAM partners.
  You can specify the selector in printable or hexadecimal format (0xnnnn...).

The specification will depend on the type of partner:

– openFT partner:
Length, 1 through 8 characters; alphanumeric characters and the special characters # @ $ are permitted. A printable selector will be coded in EBCDIC in the protocol and may be padded with spaces internally to the length of eight characters.

Default value: **$FJAM**

ssel

Session selector under which the file transfer application is accessible in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...).

Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector will be coded as variable length ASCII in the protocol.

Default value: empty

Note:
In openFT, printable session selectors are always used with uppercase characters even if they are specified or output in lowercase characters.

psel

Only relevant for FTAM partners, not used under z/OS.

Note:
In openFT, printable presentation selectors are always used with uppercase characters even if they are specified or output in lowercase characters.

Examples

The partner computer with the host name FILESERV is to be addressed over different protocols/connection types:

Connection type/protocol	**Address specification**
openFT partner	FILESERV
FTAM partner (BS2000, Windows or Unix system with default setting as of V11.0)	ftam://FILESERV
FTAM partner (Windows system with default setting up to V10.0)	ftam://FILESERV:.SNI-FTAM
Third-party FTAM partner	ftam://FILESERV:.TS0001.SES1.PSFTAM
FTP partner	ftp://FILESERV
SNA partner via openFT protocol (FILESERV is the LU name)	FILESERV:sna

3.5.2 Transfer admission

The transfer admission consists of the login name, the account number and the password (access via login/LOGON admission). These values are system-dependent. You can, however, also specify an FTAC transfer admission with an operating system-independent definition which provides a higher degree of access protection.

System	FTAC transfer admission	Login name	Account number	Password
BS2000	8 - 32 character long C string or 15 - 64 character long X string	1 - 8 alphanumeric characters	1 - 8 alphanumeric characters	1 - 32 character long C string or 1 - 16 character long X string
Unix based	8 - 32 characters long C string or 15 - 64 characters long X string	1 - 32 characters	Unix systems do not recognize any account numbers locally	Alphanumeric characters (the length is system dependent), a distinction is made between uppercase and lowercase
Windows	8 - 36 characters	1 - 36 characters, possibly with leading domain name (DOM\)	Windows does not recognize any account numbers locally	8 - 32 character long C string or 15 - 64 character long X string
z/OS	8 - 32 character long C string or 15 - 64 character long X string	1 - 8 alphanumeric characters	max. 40 characters, uppercase, digits and special characters $, @, #	1 - 8 alphanumeric characters

Examples

If you do not possess FTAC transfer admission then you can specify the transfer admission for the individual platforms using the following syntax:

- BS2000/OSD:

  ```
  userid,account-number[,'password']
  ```

- Unix systems

  ```
  userid[,,password]
  ```

- Windows systems:

  ```
  userid[,,password]
  ```

The user ID consists of a user name (In the case of local IDs, the "`host name\`" must not be entered in front of the user ID.) or, if a user ID in a LAN Manager or Windows domain is accessed, it consists of the domain name followed by an backslash (\) and the user name.

- OS/390 and z/OS:

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

  The accounting number is optional with more recent z/OS versions.

- In the case of other partner systems, your specifications depend on the conventions used in the partner system.

Inbound access using the default FTP client

If you wish to access an openFT server from a standard FTP client, you should note the following:

- Establishing a connection
  If the default listener port 21 is set on the openFT FTP server, enter the following from the shell (Unix systems), from the command prompt (Windows) or on command level (BS2000 and z/OS):

  `ftp` *hostname*

  hostname is the host name of the openFT FTP server.

  If a listener port other than 21 is set on the openFT FTP server, you need two commands to establish a connection:

  ```
  ftp
  ftp> open hostname port-number
  ```

- Login
  If you log in without an FTAC transfer admission, enter the login data interactively as usual (user ID and any password that is required and/or account number). If you log in using an FTAC transfer admission, enter the FTAC transfer admission under *User* and leave the *Password* empty.

  Example

  ```
  User: ftpuser1
  Password: (empty)
  ```

  With openFT FTP servers as of V11, you can enter the value *$ftac* under *User* and the FTAC transfer admission under *Password*.

  Example

  ```
  User: $ftac
  Password: ftpuser1
  ```

3.6 Options for file transfer

openFT offers the possibility to make additional optional setting for file transfer. You can define individual record lengths, agree syntax rules and file compression, and specify conditions for result messages.

3.6.1 Maximum record lengths

The maximum record length is understood to be the length of the longest record (net record length) not including the record length fields.

In BS2000 and z/OS files, the maximum record length is stored as a file attribute in the catalog (with variable-length records and an additional allowance of 4 for the record length field).

When transferring files from a Unix system, Windows system or POSIX (files for which there is no catalog entry specifying a maximum record length) you can set the maximum length of your file which you wish to transfer as text or record-structured binary file (user format) individually. The prescribed maximum record length must be at least as large as the largest one actually available, otherwise the FT request cannot be executed.

3.6.2 Syntax rules

With the option “Syntax rules”, you can define the procedure to be adopted for the destination file during file transfer. This option can also be defined via FTAC. There are two options:

– to overwrite files, i.e. files are overwritten, provided that the file attribute permit this action, or file that do not exist are created,

– to extend files, i.e. existing files are extended at the end of the file, provided that the file attribute permit this action, or file that do not exist are created,

– to not overwrite files; in this case, existing files are under no circumstances overwritten; rather, the FT request is aborted and an appropriate message output. If the specified destination file does not exist, a new file is created.

In z/OS, the precise effect of the "Syntax rules" option (WRITE-Mode operand) also depends on the type of receive file (PS data set, member of a PO data set, etc.). This is described in detail in section “Attributes of receive files” on page 66.

Access protection for send and receive files

Please note that the destination file is generally not protected from being overwritten by other users while the time the request is being processed. If the transfer is interrupted, for example, then other users may be able to write to the destination file. Access protection differs in the individual systems:

- openFT for BS2000 uses a file lock which protects the files if the transmission is interrupted and between the time of accepting and processing the FT request. This protection does not apply to library members and POSIX files.
- openFT for z/OS protects send and receive files against simultaneous (write) accesses only if data is in fact being transferred, i.e. if the request is in the ACTIVE state. It follows, that the send and receive files are not protected, if the file transfer has not yet begun or has just been interrupted.

  If openFT attempts to access a send or receive file which is locked (for example, because another FT job is already accessing it), the FT job is rejected or terminated.

  For a member of a PO or PDSE data set, this means:

  – When a member of a PO or PDSE data set is to be read (send file), no other member of the same data set may be open for writing or only for reading when the request is issued, nor be opened before the end of the file transfer.

  – When a member of a PO or PDSE data set is to be written (receive file), neither another member of the same data set, nor the data set itself may be open when the request is issued or opened before the end of the file transfer. In this case (receive file), even the display of the member list can cause the FT job to be aborted (for example when a send request is started from the menu interface, see page 139ff, or the use of the PDF function "member list" in general).

  When an FT request is aborted because of an attempt to access a locked file, an error message is output.
- In other systems, for example Unix and Windows systems, or even BS2000, the user is solely responsible for guaranteeing exclusive access to the files to be transferred in the case of POSIX file or library elements. In theses systems, the file cannot be exclusive openFT, not even during file transfer.

  The user him/herself must therefore ensure that (the data and file attributes) in the file to be transferred are consistent throughout the entire duration of the FT request. This applies to both the send and receive files. The danger of eventual inconsistencies resulting from multiple accesses can be reduced, for example, by means of access restrictions (Unix system: *chmod* command). It is also possible to transfer the file to a different name or to a temporary directory and to rename it or move it to a different directory only after file transfer has been completed successfully using follow-up processing.

openFT without FTAC functionality offers the same transfer and access protection as the operating system. The FT user must produce authorization for access to a file via the FT system in the same manner as for the file management system of the operating system. This means that a complete LOGON admission comprising the login name, the account number and password, as well as any file password required, must be given.

The use of openFT with FTAC functionality is an extension of the transfer and access protection features of the operating system to include the security mechanisms contained in the FTAC functionality.

The software products SYS1.UADS and RACF (or compatible products like TOP-SECRET and ACF-2) installed in the z/OS system are used to check the transfer and access admissions of the FT user. Therefore, the same conditions to read and write file access for openFT and TSO or JES2-/JES3 users.

For send files and existing receive files, openFT uses the products named above to check the FT user's access rights (read/write) against the user ID and password specified in the TRANSFER-ADMISSION and, if necessary, against the file password. If this check is negative, the file transfer is not executed, and a appropriate message.

For data protection reasons, this message gives no indication of which of the parameters USER-IDENTIFICATION, ACCOUNT or PASSWORD or the file password has been violated.

If a receive file does not yet exist, it is created by openFT. Here, too, openFT uses the products named above to check the access rights (write) against the user ID and password. If this check is negative, the file transfer is not executed, and the same message as above is output.

openFT does not assign access protection attributes to new files. In particular, no file password is given to the system and the "RACF bit" is not set in the DSCB of openFT. If you want to give immediate protection against unauthorized access to a new file under z/OS, you are recommended to use the RACF function "generic profile", which is described below.

This RACF function can be used to assign common protection to a group of files with a similar name structure. For example, all the files of a specific user ID with a name which contains the string TRANS can be protected against access by other user IDs. This also applies to files created by openFT.

The file protection attributes of the send file are not transferred to the receive system and therefore cannot be adopted for a new receive file.

openFT does not set a modification time limit for the file.

3.6.3 Compressed file transfer

Files can be sent using data compression. This shortens transmission times and saves costs. However, do note that compression and decompression produce extra CPU load in the receive processor.

openFT is able to use two compression methods - zip compression (with openFT partners as of V10) and byte compression. Both of these can be used to reduce the volume of data for transfer. However, compressing and decompressing the data increases CPU demand and consequently also the time required for a request before and after data transfer itself.

On "fast" lines (as of approximately 10 Mbit), the overall execution time of a request normally is not significantly improved by compression. On "slow" lines (less than1Mbit), zip compression may help enhance performance. Byte compression is worthwhile when transferring files which contain a large number of byte repetitions (e.g. lists with blanks for column alignment, dumps with numerous zeros). If the partner does not support compression, openFT transfers the file uncompressed. openFT-FTP supports byte compression as described in RFC959.

3.6.4 Encrypted file transfer

openFT can send data with encryption if requested by the user (see also the section “Encryption for file transfer requests” on page 44).

openFT generally uses the RSA/AES encryption procedure for request description and user data. In the case of connections to partners with older openFT versions (lower than V8.0) then the RSA/DES procedure is used for encryption.

For legal reasons, the encryption option is not available in all countries, i.e. the encrypted file transfer with foreign partners is not guaranteed in all cases.

Data encrypted by openFT van only be exchanged via the FTP protocol in an outbound direction and only with standard secure FTP partners.

Encrypted file transfer always requires openFT-CR to be installed on the openFT side, i.e. also on the partner system if openFT is running there.

3.6.5 Notifying results

The initiator of a file transfer request can arrange to be notified of the result. The logging function, which is available in a standard form on all platforms, is particularly suitable for this.

Other ways of notifying results are platform-dependent:

– In z/OS and BS2000 systems, a file is created on request by the initiator and can be printed out automatically on success or failure of the file transfer.

– In Unix systems, the result message can be stored in the mailbox of the initiator depending on the result.

3.6.5.1 Messages and return codes automatically issued by openFT for z/OS

openFT sends you automatically, i.e. without your needing to make a specific request, a message indicating acceptance or rejection of the file transfer request immediately after it has been submitted.

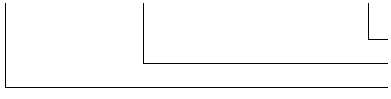
The request confirmation or rejection is issued to the TSO terminal from which the NCOPY command was entered.

At the same time, a return code is entered in system variable &LASTCC ("control variable") indicating whether the NCOPY command (**not** the file transfer request) has been successful. The return codes are described in the section “Introduction to the NCOPY command” on page 312.

If a result list is requested in the NCOPY command and this result list is automatically printed out by openFT (LISTING=*SYSLST, see the operand description for LISTING in the section “Full form of the NCOPY command” on page 316), an asynchronous message indicating termination of the print job is additionally issued to the TSO user whose user ID was specified in the TRANSFER-ADMISSION for the local system. This message is generated by means of the job parameter NOTIFY in the appropriate format.

Example of a NOTIFY message:

```
14.04.24 JOB05252 $HASP165 OPFTWITY ENDED AT P391  MAXCC=0 CN(INTERNAL)
```



Name of the print job (see below)
Job number of the print job
Current time of the print job,
format hh.mm.ss (hour.minute.second)

This message does not contain the job ID and does not indicate whether the file transfer was successful.

(This type of message is only issued if the job used to print out the result list contains a NOTIFY parameter; please consult your FT administrator.)

The FT administrator can also set openFT so that, once a transfer request is completed, an asynchronous message is issued to the TSO user whose user ID was specified in the TRANSFER-ADMISSION for the system involved.

(openFT can only issue this type of message if it is "APF authorized"; please consult your FT administrator.)

The FT administrator can specify the circumstances under which this type of message is to be output.

The FT administrator can also determine the text of these messages. Unless otherwise specified, the following default messages are issued:

– following successful file transfer:

```
FJM 2100 FILE TRANSFERRED, TRANS_ID:nnnnnnnnnn
```

– following unsuccessful file transfer:

```
FJM 2101 FILE NOT TRANSFERRED, TRANS_ID:nnnnnnnnnn
```

where "nnnnnnnnnn" is the ID of the file transfer request in the system involved.

openFT uses the TSO command SEND with the LOGON specification for issuing these messages. This means that

– the user, if already logged on to a TSO terminal, receives the message immediately. Otherwise, the user receives the message immediately after logging on.

– the user can suppress the output of these messages and other messages created by means of the SEND command (TSO command `PROFILE NOINTERCOM`) and cancel this suppression (`PROFILE INTERCOM`).

The FT administrator can also specify whether or not console traps are to be output. If console traps are output, the message FTR0340 (success) or FTR0341 (failure) is generate for each request provided that the request was entered in the request queue.

3.6.5.2 Result lists generated by openFT for z/OS

The FT system only generates a results list for an FT request if this is explicitly requested. By default, no list is printed (LISTING=*NONE) since the log records provide information about the success or failure of the request. It is therefore not essential to print a list for each request.

If a result list is required, it can be automatically output at a printer (LISTING=*SYSLST) or saved to a PS data set with the following properties (LISTING=*LISTFILE):

– Volume: the same volume as for non-existent receive files. The FT administrator can specify the volume for these (the same for all transfer requests). If the FT administrator has not specified a volume, the system defaults for newly created files apply.

– Name: <userid>.<inst>.T<transfer-id>.LST
If the "transfer-id." consists of more than 7 characters, it is split into two parts in this name as follows:
<userid>.<inst>.T<part1-id>.T<part2-id>.LST,
where "part1-id" always has a length of 7 characters and "part2-id" a length of 1 to 3 characters.
"user-id" is the user ID which was specified in the local TRANSFER-ADMISSION.

The file containing the result list is not created until the follow-up processing, if any, has been started. It is not possible, therefore, to access this file in follow-up processing.

If the result list file is to be automatically output to a printer, openFT sets up a job for this purpose, which it presents to the Internal Reader.

The FT administrator can predefine the basic structure of the job generated by openFT to print the result list. Further information is given in the System Administrator Guide "openFT for z/OS - Installation and Administration".

3.6.5.3 User-generated result information

FT users generate their own result information by specifying in the file transfer request a follow-up processing operation which outputs a message once the file transfer is complete. This type of result information can be triggered in both the local and the remote system. For example, the TSO command SEND can be used in the local system in follow-up processing.

3.6.6 Preprocessing and postprocessing

The “preprocessing” and “postprocessing” functions make it possible to execute any TSO commands (operating system commands, procedures, etc.) with the aid of a file transfer request in the local and remote systems. The commands are passed to the corresponding system instead of the file name. To do this, the file name must be specified as a C string. The first character is a pipe symbol ’|‘. Then follow the commands, separated by ';' (or '&' or '&&' in Windows systems, in which case the command string must start with *cmd /c*). The maximum length of the pre- and postprocessing command is limited by the maximum length of the file name.

If the characters ’|&’ are specified instead of the pipe symbol, the transfer request is restartable, see page 113.

Preprocessing passes the result to the system’s standard output (SYSLST on BS2000, SYSPRINT on z/OS, stdout on Unix systems and Windows systems). Postprocessing reads the data from the relevant system’s standard input (SYSIN on BS2000, SYSTSIN on z/OS, stdin on Unix systems and Windows systems).. However, the standard output/input does not usually support all the file formats possible at the system in question. You can avoid this restriction by using the %TEMPFILE variable instead of the standard output/input. This has the advantage of permitting the use of any required file format. Even if a preprocessing command cannot be output to the standard output if or a postprocessing command cannot read from standard input, normally it may be helpful to specify %TEMPFILE in the request parameters.

You should construct command sequences using the TSO WHEN command, e.g.:

```
command1;WHEN SYSRC(< 12) command12;WHEN SYSRC(< 12) command13;...
```

Pre- and postprocessing are part of the request brackets. The issuer of the request always receives a feedback report on the successful or unsuccessful completion of the pre/postprocessing.

If preprocessing or postprocessing are performed in z/OS, then the commands are started as a TSO job:

- If you specify a preprocessing command on send, then the specified commands are initially started as a TSO job. The data is output via SYSPRINT or %TEMPFILE to a temporary file that is passed to the partner (“preprocessing”). If the data is passed with SYSPRINT, you must explicitly specify OUTPUT=*STDOUT to prevent the output from being written to job logging (if SYSTSPRT DD SYSOUT=* was previously specified in the batch job) and instead write it to the specified file (or stdout, see example).

Examples

1. NSTATUS in z/OS as preprocessing for a request submitted in the Unix system

```
ncopy part!"|nstatus output=*STDOUT" file transadm
```

2. LISTCAT in z/OS as preprocessing for a request submitted in the Unix system

```
ncopy part!"|listcat ofile(sysprint)" file transadm
```

- If you specify a postprocessing command on receive, openFT supplies the first command with the transferred data via the %TEMPFILE variable or via SYSUT1 and waits until processing has been concluded (“postprocessing”).

You should note the following when using the pre/postprocessing function:

- Preprocessing/postprocessing runs as part of the file transfer operation and under the same transfer admission. The admission must contain the user ID, account and password for z/OS. These specifications are either explicitly stated in the file transfer request or in a transmission profile’s USER-ADMISSION. In the case of follow-up processing, different rights may apply depending on the platform (PROCESSING-ADMISSION).
- If the request is handled via an FTAC profile, the FILE-PROCESSING function must be permitted in the profile or, alternatively, a file name prefix starting with the pipe symbol '|' must be defined.
- In the case of preprocessing only the command’s SYSPRINT or %TEMPFILE output is transferred. The SYSPRINT or %TEMPFILE output is temporarily stored in a file which is deleted following transfer. This file is created with a unique file name in order to prevent conflicts between file processing operations that are running in parallel. The ID under which file processing is running must possess sufficient space for the creation of the temporary file as otherwise file processing will be aborted
- The temporary files that are created for pre/postprocessing are all of type "Variable Blocksize (VB)", and are automatically deleted as soon as transfer and/or preprocessing and postprocessing are completed.
- When non-restartable pre/postprocessing is involved, the connection to the partner must remain intact until the entire processing session is completed.

Restart capability during preprocessing and postprocessing

During restartable pre- and postprocessing, the data to be transferred between openFT and the processing command is always saved to a temporary file. By this means, the request is divided into 3 phases: preprocessing, transfer, and postprocessing.

The restart capability of a pre- and postprocessing session is brought about when you specify an additional “&” before pre- and postprocessing in the transfer command. During this, requests made with openFT partners behave as follows:

- Loss of connection during preprocessing:
  If the connection is lost during the execution of the preprocessing command, the command is still executed until completion after the connection is lost. If the system is restarted after the command has completed execution, then the temporary file is transferred.
- Loss of connection during transmission:
  In this case openFT performs a restart for the temporary file as is usually the case.
- Loss of connection during postprocessing:
  If the connection is lost during the execution of the postprocessing command, the command is still executed until completion after the connection is lost. If the system is restarted, then all other actions left over that belong to the openFT request are performed (e.g. any follow-up processing or the status report to the partner).

Server function for remote command execution

One special form of preprocessing is the server function for the remote command execution (FTEXEC command). This command makes it possible to execute commands on a remote system. The exit code and/or the output from *stdout* and *stderr* (Unix or Windows systems), SYSLST and SYSOUT (BS2000) or STDOUT=SYSPRINT und STDERR=SYSTSPR (z/OS) are output at the local computer. FTEXEC thus mimics the execution of the command on the local computer.

If ftexec is used at a Windows or Unix system for the remote execution of z/OS commands, then the command’s SYSPRINT output is routed to stdout and the SYSTSPRT output to stderr. On z/OS, the openFT commands supply the return codes (0 successful; 4 warning; 8 currently reserved; 12 error). However, the FTEXEC server only distinguishes between success (exit code 0) and failure (exit code 12). It also interprets return code 4 as indicating success and maps this to 0.

Preprocessing and postprocessing jobs generated by openFT

If you specify a TSO command (or a sequence of TSO commands) for preprocessing or postprocessing, openFT generates a job that is responsible for processing. The FT administrator can predefine the basic structure of the job. This is done in the elements TSOVVJOB, TSONVJOB and TSOVFJOB of the openFT parameter library. For more information see the System Administrator Guide "openFT for z/OS - Installation and Administration".

3.6.7 Follow-up processing

openFT offers four types of follow-up processing requests:

– Follow-up processing in the local system after successful file transfer
– Follow-up processing in the remote system after successful file transfer
– Follow-up processing in the local system after unsuccessful file transfer
– Follow-up processing in the remote system after unsuccessful file transfer

The conventions of the system on which the follow-up processing is to be performed are decisive for the syntax and processing of the statements and commands. A command sequence can only be processed in the remote system if an FT that supports this function in used in the remote system.

You may specify variables within the command or command sequence for follow-up processing. These are substituted at the start of follow-up processing in the particular system using the values obtained from the file transfer requests. The following table shows which variables can be used for which system.

Variable	Meaning	BS2000	Unix system	Windows	z/OS
%PARTNER	Partner name (long form)	X	X	X	X
%PARTNERAT	Partner name (short form)	X	X	X	X
%FILENAME	File name	X	X	X	X
%ELEMNAME	Element name	X			
%ELEMVERS	Element version	X			
%ELEMTYP	Element type	X			
%RESULT	Request result	X	X	X	X
%JOBCLASS	Job class	X			

In the case of %PARTNER and %PARTNERAT, the partner name found in the partner list is used if it is present in the partner list. If it is not entered in the partner list (dynamic partner) then the partner address is used. In this case, %PARTNER and %PARTNERAT have different effects:

– In the case of %PARTNER, all the address components are used, i.e. including protocol prefix, port number and selectors if appropriate.

– In the case of %PARTNERAT, only the *host* address component is used, see page 100. In addition, all characters apart from letters, digits or periods are replaced by '@'.

Further variables you can specify in the commands or the command sequence for follow-up processing for an z/OS system are described on page 116.

You may specify data for follow-up processing both for the local and for the remote system, depending on the version of openFT-Version used. In each case, no more than 1000 characters may be used. The number of characters evaluated depends on the operating system and is stated in the relevant FT description. Please observe that

- the limit length applies after any necessary translation of variables.
- as of openFT V12, follow-up processing commands in Windows systems are converted into the UTF-8 character code and that therefore characters that are not present in the ISO646 character set occupy more than one byte in memory.

The limit of up to 1000 characters can be bypassed by calling a procedure, a shell script or a program from within the follow-up processing. A procedure may contain the command sequence which is to be executed on success or failure of file transfer.

Restrictions apply to links with FTP partners, since the FTP protocol does not permit transfer of follow-up processing data. Follow-up processing in the FTP partner system is possible only if it is stipulated there in an FTAC admission profile. It is always possible to initiate follow-up processing in the local system.

The special form of follow-up processing, *DELETE , is available for requests on which the send file is to be deleted following successful transmission, This character string can be specified as follows:

– as remote follow-up processing for synchronous and asynchronous receive requests,

– as local follow-up processing for asynchronous send requests or with FTP partners.

*DELETE causes openFT itself to delete the sent file in the sending system after the termination of the FT request without it being necessary to start a batch job. However, as in the case of "genuine" follow-up processing that consists of system commands, *DELETE does not form part of the job scope. This means there is no response message indicating whether or not the file has been successfully deleted. “Genuine” follow-up processing can be additionally specified via an FTAC profile.

To avoid undefined file fragments in the event of unsuccessful file transfer, it is useful to delete the receive file via follow-up processing in such cases.

Jobs created by openFT for follow-up processing

If you specify a TSO command (or a sequence of TSO commands) or a JCL statement (or a sequence of JCL statements) for follow-up processing, openFT creates a job for the execution of follow-up processing. The FT administrator predefines the basic structure of these jobs. This is done using the elements TSOJOB and/or JCLJOB in the FT parameter library. Further information is given in the System Administrator Guide "openFT for z/OS - Installation and Administration".

A special case of follow-up processing is initiated in openFT for z/OS by means of the character string ALLOC DSNAME. This specific openFT statement is used to specify the name of a cataloged PS data set or of a member of a cataloged PO or PDSE data set containing a complete executable MVS job. (In this case, openFT does **not** support the use of relative generation data sets.)

This job is started by openFT within follow-up processing via the Internal Reader. openFT does **not** generate any additional job control statements in this case, thus permitting follow-up processing jobs with user-specific job parameters to be executed.

The records in the data set/member must have a fixed length of 80 (LRECL=80, RECFM=F or =FB). The name of the PS, PO or PDSE data set may be specified in the openFT statement ALLOC DSNAME by its fully qualified name (i.e. with "first level qualifier") and must be enclosed in single quotes. (Since the character string for follow-up processing is also enclosed in single quotes, a fully qualified file name must be enclosed in two sets of single quotes.) If this is not the case, the user ID from the PROCESSING-ADMISSION is prefixed to the file name as the "first level qualifier" by openFT.

For more details about the syntax of this specific openFT statement, see the operand description for SUCCESS-/FAILURE-PROCESSING in the section “Full form of the NCOPY command” on page 316.

Using variables

You can use variables for follow-up processing in z/OS:

– in the TSO commands and JCL statements specified in the NCOPY command,
– in the job envelope specified for follow-up processing by the FT administrator and
– in the statements in the jobs that are to be started as follow-up processing with the openFT statement ALLOC DSNAME

openFT replaces the variables with the current values before starting follow-up processing.

The following variables can be used:

FILENAME	Name of the send or receive file. Maximum 56 characters in accordance with openFT conventions. This variable can only be used in the NCOPY command and must start with %.
FILN	Name of the send file or receive file of the relevant system, as specified in the NCOPY command. A maximum of 56 characters in accordance with openFT conventions.
FILX	Like FILN, except that quotes are automatically doubled during the replacement.
PARTNER	Symbolic name of the partner system. Maximum 8 characters. This variable can only be used in the NCOPY command and must start with %.
PNAM	Symbolic name of the partner system. A maximum of 8 characters.
RESULT	Four-digit return code indicating the result of file transfer. It corresponds to the reason code in the FT log record or the message code of the corresponding FTR message. This variable can only be used in the NCOPY command and must start with %.
USID	User ID from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 7 characters in accordance with IBM conventions.
ACCN	Accounting information from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 40 characters in accordance with IBM conventions.
ACCX	Like ACCN, except that quotes are automatically doubled during the replacement.
PASS	Password from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 8 characters in accordance with IBM conventions.
OWID	Owner of the FT job, i.e. the user ID under which the transfer job was created. A maximum of 7 characters in accordance with IBM conventions. This parameter is replaced only in the system in which the transfer job was created. It is eliminated in the partner system. It should therefore be specified only in follow-up processing for the local system.
PGRN	Programmer's name, as specified with the keyword PGRN= as a subcommand in the command string for follow-up processing in the relevant system (see section “Specifying a programmer's name” on page 122). A maximum of 20 characters in accordance with IBM conventions.

PGRX Like PGRN, except that quotes are automatically doubled during the replacement.

TRID FT job identifier in the relevant system. A maximum of 10 characters (value range 1..2147483639) in accordance with openFT conventions.

The names of these variables must be prefixed with the character "%" or - for reasons of compatibility with previous versions - "&". In follow-up processing for remote systems of the type openFT for z/OS (specification REMOTE=*MSP in the NCOPY command), the character "&" should always be used to make sure that previous versions also handle the variables correctly.

The names of these variables should have as many trailing "#" fill characters as are necessary for a field to be set to its maximum length (including the "&" character, e.g. &TRID#####). When replacing the variables by the current values, openFT does not exceed the field length predefined by the name of the symbolic parameter including the trailing "#" fill characters; if necessary the current values are truncated. On the other hand, where the current values are shorter than this field length, openFT removes superfluous blanks.
For the FILX, ACCX and PGRX parameters, in which quotes are automatically doubled during the replacement, the number of trailing characters must be increased by the number of quotes possibly contained in the current value (i.e. 2 additional trailing characters for FILX and ACCX, whose current value can be enclosed in quotes, but which cannot themselves contain quotes, and at least 3 additional trailing characters for PGRX for the external quotes and at least one quote contained in the name).

Example 1

The following transfer job is requested by user USER0 in system SYS1:

```
NCOPY TRANS=TO, +
      PARTNER=SYS2, +
      LOC=(FILE='TEST1.ABC', +
           TRANS=(USER1,ACC1,PASS1), +
           PROC=(USER3,ACC3,PASS3), +
           SUCC='//STEP     EXEC PGM=IEFBR14 CALL PROGRAM; +
                 //DELFILE  DD DSNAME=&FILX#########################+
##########################,DISP=(OLD,DELETE,DELETE); +
                 //* PGRN=''MAC''''DONALD''', +
           FAIL='SEND ''job &TRID##### with USERID=&USID## not +
                 correct.'', USER(&OWID##); +
                 //* PGRN=''MAC''''DONALD'''), +
      REM=*MSP(FILE=FROM.USER1, +
               TRANS=(USER2,ACC2,PASS2), +
               PROC=(USER4,ACC4,PASS4), +
               SUCC='SEND ''FILE &FILX############################+
```

```
######################## from &PNAM### received.'', USER(&USID##); +
                       //* PGRN=DONALD', +
                  FAIL='SEND ''job &TRID##### from partner &PNAM### +
                       not correct.'', USER(&USID##); +
                       //* PGRN=DONALD')
```

If the standard jobs are used, openFT creates the following jobs for follow-up processing (for help with the PGRN parameter:

If the job is successful in the local system:

```
//USER3N    JOB ACC3,
//           'MAC''DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//           NOTIFY=USER3,
//           USER=USER3,
//           PASSWORD=PASS3,REGION=0M
//           EXEC PGM=IKJEFT01
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
//STEP EXEC PGM=IEFBR14
//* CALL PROGRAM
//DELFILE DD DSNAME='TEST1.ABC',DISP=(OLD,DELETE,DELETE)
/*
```

If the job is unsuccessful in the local system:

```
//USER3N    JOB ACC3,
//           'MAC''DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//           NOTIFY=USER3,
//           USER=USER3,
//           PASSWORD=PASS3,REGION=0M
//           EXEC PGM=IKJEFT01
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
SEND 'request 1234567890 with USERID=USER3 not correctt.', USER(USER0)
/*
```

If the job is successful in the remote system:

```
//USER4N    JOB ACC4,
//           'DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//           NOTIFY=USER4,
//           USER=USER4,
//           PASSWORD=PASS4,
//          EXEC PGM=IKJEFT01,REGION=512K,DYNAMNBR=10
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
SEND 'File FROM.USER1 received from SYS1 .', USER(USER4)
/*
```

If the job is unsuccessful in the remote system:

```
//USER4N    JOB ACC4,
//           'DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//           NOTIFY=USER4,
//           USER=USER4,
//           PASSWORD=PASS4,REGION=0M
//           EXEC PGM=IKJEFT01
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
SEND 'Request 0987654321 from partner SYS1 not correct.', USER(USER4)
/*
```

Example 2

USER1 in system SYS1 sends an invoice to USER2 in system SYS2 with the following batch job. If the file transfer is successful, USER1 is to receive an acknowledgement automatically (via file transfer from SYS2 to SYS1).

```
//TESTJOB   JOB
//          EXEC PGM=IKJEFT01
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
NCOPY TRANS=TO, PARTNER=SYS2,                                                -
LOC=(FILE=INVOICE,                                                           -
     TRANS=(USER1,ACC1,PASS1),                                               -
     LIST=*NONE),                                                            -
REM=*MSP(FILE=INVOICE.SYS1,                                                  -
         TRANS=(USER2,ACC2,PASS2),                                           -
         PROC=(USER4,ACC4,PASS4),                                            -
         SUCC='NCOPY TRANS=TO,PARTNER=&PNAM###,                              -
LOC=(FILE=ACKNOWLEDGMENT,TRANS=(&USID##,&ACCN##############,&PASS###)),      -
         REM=*MSP(FILE=QUIT.SYS2,TRANS=(&USID##,&ACCN#########,&PASS###))'),-
WRITE=*REPLACE
/*
//
```

If the standard jobs are used, the openFT for z/OS installed on the remote system (SYS2) creates the following job for follow-up processing:

```
//USER4N    JOB ACC4,
//USER4N    JOB ACC4,
//          'DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//          NOTIFY=USER4,
//          USER=USER4,
//          PASSWORD=PASS4,REGION=0M
//          EXEC PGM=IKJEFT01
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
NCOPY TRANS=TO,PARTNER=SYS1,LOC=(FILE=ACKNOWLEDGMENT,
TRANS=(USER4,ACC4,PASS4)),REM=*MSP(FILE=QUIT.SYS2,TRANS=(USER4,ACC4,PASS4))
/*
```

Specifying a programmer's name

The programmer's name can be specified in the command string for follow-up processing for the local system and for the partner system, if this is also an openFT V5.0 for z/OS. A separate value for the programmer's name can be specified in each of the four command strings for follow-up processing provided in the NCOPY command (local system and partner system, both for successful and unsuccessful transfers). openFT uses the specified programmer's name in the following locations:

- in the JOB statement of the jobs for follow-up processing; the following applies here:
  - this specification is used automatically in the jobs that openFT creates by default when the computing center does not provide its own defaults
  - this specification can also be used in jobs for follow-up processing predefined by the computing center; this is described in more detail in the System Administrator Guide "openFT for z/OS - Installation and Administration"
- in the JOB statement of the job for printing out the result list; the following applies here:
  - this specification is used automatically in the job that openFT creates by default when the computing center does not provide its own defaults
  - this specification can also be used in the job predefined by the computing center for printing the result list; this is described in more detail in the System Administrator Guide "openFT for z/OS - Installation and Administration"
- as a substitute value for the variable PGRN or PGRX, which can be used in the statements for follow-up processing (see page 116)
- as a substitute value for the variable PGRN or PGRX, which can be used in the message texts predefined by the computing center for asynchronous messages once a file transfer has been completed; this is described in more detail in the "System Administrator Guide "openFT for z/OS - Installation and Administration"

The programmer's name must be specified as a subcommand within the command string for follow-up processing with the keyword PGRN=. The following additional conditions must be observed:

- The subcommand "PGRN=value" may be used only once within a string. If this is not the case, the analysis is terminated, i.e. any follow-up processing is not started, and no value for the programmer's name is saved for further use.
- The "value" specification must come directly after the character "=" and ends with the first blank or, if it is within quotes, with the last quote in the substring. There must be no other character except the blank between the end of the value determined in this way and the end of the substring (";" or end of string); otherwise, the analysis is aborted (see above).

- Each quote which occurs must be doubled, since follow-up processing strings are themselves enclosed in quotes (c string).
- The value specifying the programmer's name must comply with the following IBM syntax rules:
  - Maximum length 20 characters, excluding enclosing quotes
  - The specification must be enclosed in quotes when metacharacters are used; exception: hyphen, leading periods or embedded periods.
  - Each quote which is part of the name must be doubled.
- If the partner system is not an openFT for z/OS (specification REMOTE= *BS2000/*ANY in the NCOPY command), no specification may be made for the programmer's name in the command string for follow-up processing in the partner system, since these partner systems do not support the specification of a programmer's name.

Of these rules, openFT checks only the length of the value specified for the programmer's name. If it is longer then the maximum permissible length, the analysis is aborted (see above).

Specifications for the programmer's name which have the wrong syntax, and particularly the wrong number of quotes, cause an error in jobs to which the programmer's name has been assigned.

Examples

The following specifications have a valid syntax (the string "...;" here represents any sequence of TSO commands or JCL statements):

```
SUCC='...; //* PGRN=TEST-1'                          ------> (1)
SUCC='...; //* PGRN=.TEST'                           ------> (1)
SUCC='...; //* PGRN=TEST.1'                          ------> (1)
FAIL='...; //* PGRN=''TEST 1'''                      ------> (2)
FAIL='...; //* PGRN=''TEST/1'''                      ------> (2)
FAIL='...; //* PGRN=''TEST1.'''                      ------> (2)
FAIL='...; //* PGRN=''O''''REILLY'''                 ------> (3)
```

Explanations:

(1) Hyphens, leading periods and embedded periods as part of the programmer's name do not require quotes.

(2) If there are metacharacters in the specification of the programmer's name (here: blank, slash, period at the end), the specification must be enclosed in quotes. Since the follow-up string is also enclosed in quotes, each of these quotation marks must be doubled.

(3) Even when the specification of the programmer's name contains a quotation mark (here: O'REILLY), the specification must be enclosed in quotes; the quotation mark itself must be doubled (i.e. 'O''REILLY').
Since the follow-up string is also enclosed in quotes, each of these quotation marks must be doubled.

3.6.8 Accounting of file transfer requests

In z/OS, a distinction must be made between an account number and accounting information.

An **account number** may not contain blanks, tabs, double quotes, single quotes, semicolons or line control characters; brackets are permitted only in pairs.

If an account number contains only letters, digits and the metacharacters @, $, #, it can be specified without single quotes.

If it contains additional metacharacters (which must comply with the restrictions named), it must be enclosed within quotes.

Examples

– The account number A123$4 can be specified without quotes.

– The account number A(123) must be entered as 'A(123)' (this means, for example, as c'"A(123)"' in the NCOPY command, determined by the data type *c-string*).

Accounting information consists of an account number (string before the first comma), which must comply with the syntax specified above, and additional specifications with any number of characters. openFT checks only the string before the first comma for validity with the SYS1.UADS or with RACF.

Accounting information must be specified in accordance with IBM-JCL conventions. Basically, it is necessary to ensure that quotes in the accounting information are enclosed when they are received and that the quotes are not evaluated - and deleted - when the command is entered.

If the account number (string before the first comma) contains metacharacters other than @, $, #, the entire accounting information must be specified in triple quotes.

Example

The accounting information 123$#@,ABC,12/90 can be specified as '123$#@,ABC,12/90' in accordance with IBM-JCL conventions.

– At the menu interface of openFT for z/OS and the graphical user interface of openFT for BS2000, you enter this accounting information as '123$#@,ABC,12/90'.
– In an FT request issued in z/OS or BS2000, the specification is c"'123$#@,ABC,12/90"'
– At the graphical user interface of openFT for Windows or for Unix systems:
'123$#@\,ABC\,12/90'
– In a command issued under Windows, enter:
'123$#@\,ABC\,12/90'
– In a Unix system, the following must be specified in the command:
\"123$#@\,ABC\,12/90'\'

When you issue an FT request, you initiate three processes in the local and remote system that are accountable:

– initiation of file transfer by means of the NCOPY command,
– the file transfer itself, and
– any follow-up processing performed.

Initiation of file transfer by means of the NCOPY command is implemented in an interactive or a batch job accounted as usual.

Follow-up processing in the local and remote systems is charged by FT systems to the accounts whose account numbers are specified in the local and remote PROCESSING-ADMISSION of the file transfer request.

openFT for z/OS also enables the file transfer itself to be accounted. If the FT administrator sets openFT accordingly, an accounting record is written to the SMF file for each transfer request accepted, provided that SMF (System Management Facilities) is active. This applies to transfer requests issued in the local system and in remote systems. The computer center can evaluate these accounting records using accounting programs. Accounting records can only be written if the openFT is "APF authorized"; please consult your FT administrator.
The structure of the accounting records is described in the System Administrator Guide "openFT for z/OS - Installation and Administration".

Default account number

Both the actual file transfer and the follow-up processing can be charged to the default account number of the user ID which is specified in the TRANSFER-ADMISSION or PROCESSING-ADMISSION. For this to be possible, the following requirements must be satisfied:

– No account number is specified in the TRANSFER-ADMISSION or PROCESSING-ADMISSION.
– The RACF database must contain a default account number for the user ID specified in the TRANSFER-ADMISSION or PROCESSING-ADMISSION (this can be recognized, for example, by the fact that this account number appears in the TSO welcome screen). This account number may not exceed 40 characters.
– The RACF resource class ACCTNUM is active.
– The SYS1.UADS data set is not available.

For information on the last two points, please contact your FT administrator or z/OS system administrator.

3.7 File management

In particular, users of other openFT systems (clients) can make file management requests for processing files in an z/OS system.

As of V10, file management requests can also be started from z/OS (FTDEL, FTMOD, FTSHW, FTCREDIR, FTMODDIR, FTDELDIR).

This means, for example, that z/OS files can be managed with openFT for Windows from a PC, using a user-friendly interface similar to that of Windows standard. PC users can therefore fetch z/OS files to the PC (or send files from the PC to the z/OS) by dragging and dropping, without having to be familiar with the details of the z/OS file name syntax.

This presents no danger to the security of the z/OS system and the files residing there. It is also the case with file management jobs that openFT checks the user's right to access the system on which openFT is running, as well as the user's right to access the file for which the file management function is to be carried out. To do this, it uses the SYS1.UADS and RACF software products installed on the z/OS system (or compatible products, such as TOP-SECRET and ACF-2). If an openFT-specific exit routine is installed, this is also called to check file management requests.
The FTAC function is also available for extended access and data access control especially for file transfer and file management (see page 38).
(An openFT-specific Exit routine is not called to check the file management requests.)

As a file management server, openFT for z/OS supports the following functions:

– naming files
– deleting files
– querying file attributes, e.g. the size of a file
– showing directories
– naming directories
– deleting directories

In the file management functions, openFT regards both a complete z/OS file (PS data set, VSAM data set, PO data set, etc.) and a single member of a PO or PDSE data set as a **file**. Relative file generations are not supported in file management. For file management of VSAM files, as with file transfer, an ICF or VSAM catalog must be available.

openFT for z/OS handles the following as **directories**:

– all files with a common start to their name up to a qualification delimiter (period); this type of directory is referred to below as a period directory. Although directories of this type can be specified as directory names in order to display the directory contents (ftshw -d or SHOW-REMOTE-DIRECTORY), they never appear as “subdirectories” in the resulting display.

– the contents of a PO data set or PDSE data set, i.e. the members it contains; this type of directory is referred to below as a library directory. Directories of this type can be displayed, renamed and deleted. They also appear in the output directory lists.

– openEdition directories

The HOME directory of a user ID consists of all files whose names begin with this user ID (and a period).

The following section uses an example to explain how z/OS files can be managed from a PC with openFT for Windows.

This is followed (as of page 132) by a detailed description of how the various file management functions are executed by openFT for z/OS as the server.

3.7.1 File management of z/OS files with openFT for Windows

z/OS files can be managed from a PC with openFT for Windows by means of a user-friendly interface similar to that of Windows standard. This allows PC users to fetch z/OS files to the PC (or send files from the PC to the z/OS) by dragging and dropping, without having to be familiar with the details of the z/OS file name syntax. The following example will help to explain how this works.

There are the following catalog entries for user USER1 on the z/OS system:

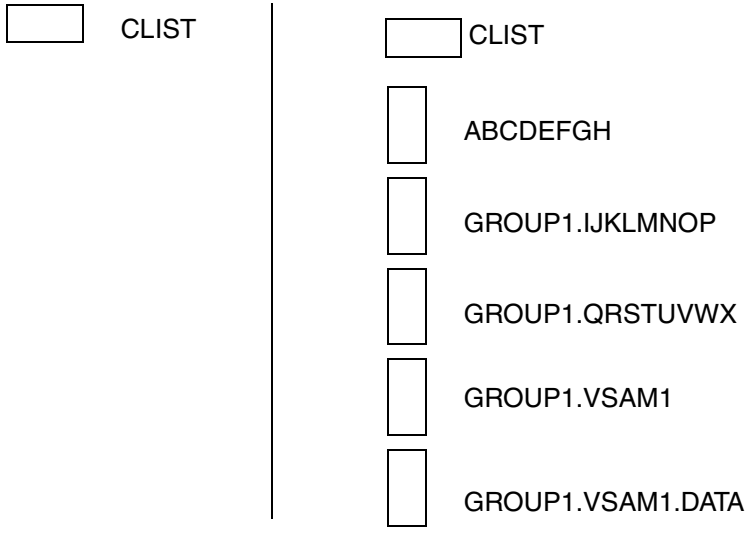
```
'USER1.ABCDEFGH'
'USER1.CLIST'
'USER1.GROUP1.IJKLMNOP'
'USER1.GROUP1.QRSTUVWX'
'USER1.GROUP1.VSAM1'
'USER1.GROUP1.VSAM1.DATA'
```

The catalog entries 'USER1.ABCDEFGH', 'USER1.GROUP1.IJKLMNOP' and 'USER1.GROUP1.QRSTUVWX' are PS data sets. 'USER1.CLIST' is a PO data set or a PDSE data set (the exact file structure is not relevant here). 'USER1.GROUP1.VSAM1' is the main entry for a VSAM cluster of the type "entry sequenced", and 'USER1.GROUP1.VSAM1.DATA' is the entry for the related data part.

If the PC user opens the HOME directory of user USER1 as a remote directory in openFT for Windows (i.e. the user enters only a period for the remote directory name), the view of these z/OS files shown below is displayed.

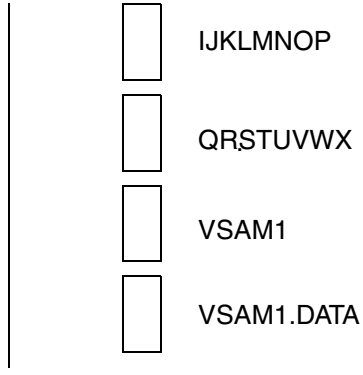
Note on the following diagrams:

Horizontal boxes represent directories, vertical boxes are files.



The PS file ABCDEFGH, the files GROUP1.IJKLMNOP, GROUP1.QRSTUVWX, and the main entry of the VSAM cluster GROUP1.VSAM1 can be renamed or deleted directly with the relevant openFT for Windows functions. It can also be dragged and dropped on to the PC.

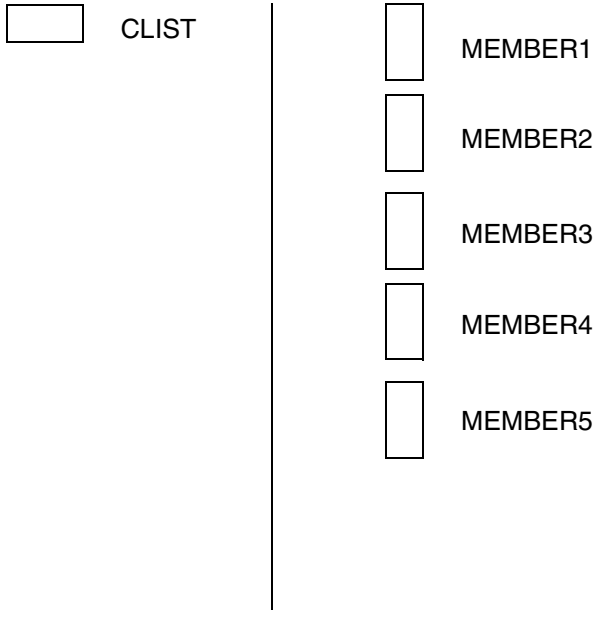
If the user views the contents of the period directory GROUP1., the following display is shown:



The PS files IJKLMNOP and QRSTUVWX can be renamed or deleted directly with the relevant openFT for Windows functions. They can also be dragged and dropped on to the PC.

The VSAM cluster can also be renamed or deleted directly with the relevant openFT for Windows functions and its contents can be dragged and dropped on to the PC. However, the main entry (VSAM1) must be used; if the PC user tries to rename or delete the entry for the data part (VSAM1.DATA) or to fetch it to the PC, the error message "Access to 'VSAM1.DATA' is denied" is output.

When the user switches from the HOME directory to the library directory CLIST, the following display is shown:



Members MEMBER1, MEMBER2 and so on can be renamed or deleted directly with the relevant openFT for Windows functions. They can also be dragged and dropped on to the PC.

3.7.2 Displaying file attributes

Users from openFT partner systems can display the attributes of z/OS files.

This can be done using the FTSHW command in z/OS.

To use this function, the user ID specified in the TRANSFER-ADMISSION does not need any special access rights for the file whose attributes are to be displayed (the "a" bit is always set by openFT for z/OS, see below). The name of the file or member to be displayed is expected in valid z/OS syntax; no type identifiers (e.g. :L:) may be used as prefixes. The name can be partially or fully qualified. If it is partially qualified, the user ID specified in the TRANSFER-ADMISSION is extended as a "first level qualifier". The following specifications therefore identify the same file:

'USER1.ABC' - with any user ID in the TRANSFER-ADMISSION

ABC - with user ID USER1 in the TRANSFER-ADMISSION)

Note that in the partner systems a fully-qualified file name must be specified as a <c string>. In this case, single quotes form part of the file name and must be passed to openFT for z/OS. For this reason, the quotes must be appropriately hidden in the request (e.g. by a \ in a shell of a Unix system or as triple quotes in BS2000). The file management commands for displaying the above-mentioned files could therefore read as follows:

BS2000:

```
SHOW-REMOTE-FILE-ATTRIBUTES PARTNER=ZOS1,FILE=c'''USER1.ABC''',
TRANSFER-ADMISSION=...
```

or

```
SHOW-REMOTE-FILE-ATTRIBUTES PARTNER=ZOS1,FILE=ABC,
TRANSFER-ADMISSION=(USER1,...),...
```

z/OS:

```
FTSHW PARTNER=ZOS1,FILE==c'''USER1.ABC''',TRANSFER-ADMISSION=...
```

or

```
FTSHW PARTNER=ZOS1,FILE=ABC,TRANSFER-ADMISSION=(USER1,...),...
```

Unix system:

```
ftshw zos1!\'user1.abc\’ trans-adm ...
```

or

```
ftshw zos1!abc user1,...
```

Windows system:

```
ftshw zos1!'user1.abc’ trans-adm ...
```

or

```
ftshw zos1!abc user1,...
```

The following section explains which file attributes of z/OS files can be displayed. The scope of information actually displayed on the client system depends on the selection made by the user in the file management command. The file attributes described here also apply when the z/OS file is shown as part of a directory (see section “Displaying directories” on page 135).

File name

This contains the file name in the following form:

– When file attributes of a single file are displayed, the name is output as entered by the user in the command (default behavior of the client systems).

– The names of the files contained in a directory (see section “Displaying directories” on page 135) are always output in relation to this directory, i.e. for a period directory only the rest of the file name after the common file name beginning is output; for a library directory only the member name is output.

Time last used

The file creation date is output. In the case of a PO or PDSE member, the library creation date is output. For reasons of compatibility, openFT always sets the time to12:00 Greenwich Mean Time since the actual time of file creation is not catalogued in z/OS.

File type

No distinction is made between text and binary files. (In the standard output of the client systems, a "*" is therefore shown in front of the access rights to indicate "No information about file structure available".)

Record format

This is shown with the following meanings:
"v" for the z/OS record format V (records of variable length)
"f" for the z/OS record format F (records of fixed length)
"u" for the z/OS record format U (undefined record format)

Maximum record length

This is shown; for files with an undefined record format (U), the block length is displayed.

Access rights

These are shown as follows:

- If RACF (or a compatible product such as ACF-2 or TOP-SECRET) is installed and active, the access rights for the user ID specified in the TRANSFER-ADMISSION are shown. They are derived from the RACF protection of the file as follows:

Access rights to the file as per RACF	Access rights shown for the file management function "Show file attributes"
NONE	-----a-----
READ	r----a-----
UPDATE	r-pxea-----
CONTROL	r-pxea-----
ALTER	r-pxeacd---

  The display "r-pxeacd---" therefore represents full access rights to the file; these access rights are shown in the following cases, as well as for ALTER:

  – RACF (or a compatible product) is installed and active, but the file is not RACF-protected.

  – openFT has RUNMODE=N (no access checks are made in this test mode).

- If RACF (or a compatible product) is not active, all access rights are shown as given (i.e. also "r-pxeacd---"). Any protection of the file by a password is ignored. However, for security reasons, actual access to the file via file management (renaming or deleting the file, see section “Renaming files” on page 136 and section “Deleting files” on page 137) is not permitted.

File size in bytes

This is shown if a reasonable value can be determined; otherwise, 0 is shown. The value 0 can therefore signify that:

– the file is empty

– no reasonable size value could be determined; this is always the case with members of PO/PDSE files.

3.7.3 Displaying directories

Users of openFT partner systems can display a directory in z/OS.

This can be done using the command FTSHW <partner>,*DIR(<dirname>) in z/OS.

The meaning of the word "directory" for openFT for z/OS is explained on page 128.

The attributes displayed for the files contained in a directory are described in section “Displaying file attributes” on page 132.

In addition to the files, libraries are also displayed as subdirectories.

Library subdirectories of the current directory are also marked as such by the "Show directories" function (in contrast to the "Show file attributes" function, in which they are handled as files; see section “Displaying file attributes” on page 132); in other words, a "d" appears in front of the attributes in the standard output of the client systems.

For this function, the user ID specified in the TRANSFER-ADMISSION does not need any special access rights for the files whose attributes are to be shown (the "a" bit is always set by openFT, see section “Displaying file attributes” on page 132).

The name of a directory can be partially or fully qualified. If it is partially qualified, the user ID specified in the TRANSFER-ADMISSION is extended as a "first level qualifier".

With regard to entering fully-qualified directory names in the partner systems, the same applies as for the name of an individual file (see section “Displaying file attributes” on page 132).

The following examples of openFT for z/OS file management commands will explain the possible specifications:

```
FTSHW PARTNER=ZOS1,FILE=*DIR(c'''USER1.TEST.'''),TRANS-ADM=...
```

All files whose "first level qualifier" is USER1 and whose "second level qualifier" is TEST will be displayed. The same can be achieved with the following command:

```
FTSHW PARTNER=ZOS1,FILE=*DIR('TEST.'),TRANS-ADM=(USER1,...),...
```

All files with the same "first level qualifier" (USER1) can be displayed with one of the following commands:

```
FTSHW PARTNER=ZOS1,FILE=*DIR(c'''USER1.'''),TRANS-ADM=...
```

```
FTSHW PARTNER=ZOS1,FILE=*DIR('.'),TRANS-ADM=(USER1,...),...
```

All members of the PO file 'USER1.TEST.LOAD' (including their alias names) can be displayed with one of the following commands:

```
FTSHW PARTNER=ZOS1,FILE=*DIR(c'''USER1.TEST.LOAD'''),TRANS-ADM=...
```

```
FTSHW PARTNER=ZOS1,FILE=*DIR('TEST.LOAD'),TRANS-ADM=(USER1,...),...
```

3.7.4 Renaming files

Users of openFT partner systems can rename z/OS files or PO/PDSE members. The FTMOD command can be used in z/OS. A file or a member with the specified new name may not already exist.

It is only possible to rename files if RACF (or a compatible product such as ACF-2 or TOP-SECRET) is installed and active. In addition, the user ID specified in the TRANSFER-ADMISSION requires the following access rights:

– The "c" bit for the file to be renamed (old name) must be set in the display of the file attributes (see section “Displaying file attributes” on page 132). This is the case when the user ID has the RACF access right ALTER or the file is not RACF-protected.

– The user ID must be authorized to create a file with the specified new name. (If there is a "generic profile" which records the new file name, it must have ALTER access.)

When openFT establishes on checking these access rights that the user may not rename the file, no attempt is made to rename the file; the request is rejected with an error message indicating that the user has insufficient access rights, in a client system with openFT as of V10 for z/OS, e.g.

```
FTR2195 REMOTE SYSTEM: OPENFT IS NOT LONGER AUTHORIZED TO EXECUTE REQUESTS
FOR THIS USER.
```

If it is the actual attempt to rename the file which fails, a different error message is output, in a client system with openFT as of V10 for z/OS, e.g.

```
FTR2018 ATTRIBUTES COULD NOT BE MODIFIED
```

There are different situations in which RACF (or a compatible product) rejects actual renaming of the file. However, this behavior can depend on the rights of the user ID under which openFT is running (as a job or as a started task), and also on the installation-specific settings for RACF behavior. The old name and the new name of the file or of the member can be specified in the same form as described for displaying the attributes of an individual file (see section “Displaying file attributes” on page 132).

For the catalog entries for a VSAM cluster, the function works as follows:

- If the name of the main entry is specified in the command as an old name, only the name of this main entry is modified. The names of the data and index parts are not changed, nor are the internal references to these catalog entries.
  (The cluster therefore remains consistent).
- If the name of the data or index part is specified in the command as an old name, the relevant catalog entry and the related internal reference are changed.
  (Again, the cluster therefore remains consistent.)
- The name of the data or index part cannot be changed.

3.7.5 Renaming directories

Users of openFT partner systems can rename z/OS directories. This includes PO data sets, PDSE data sets and openEdition directories.

To do this, the FTMODDIR command can be used in z/OS.

3.7.6 Deleting files

Users of openFT partner systems can delete z/OS files.

For this function, the user ID specified in the TRANSFER-ADMISSION requires the following access rights:

- If RACF (or a compatible product, such as ACF-2 or TOP-SECRET) is installed and active:

  The "d" bit for the file to be deleted - or the PO/PDSE file in which a member is to be deleted - must be set in the display of the file attributes (see section “Displaying file attributes” on page 132). This is the case when the user ID has the RACF access right ALTER or the file is not RACF-protected.

- If RACF (or a compatible product) is not active:
  In this case, the "Delete files" function is not permitted for security reasons.

The name of the file or of the member to be deleted can be specified in the same form as described for displaying the attributes of an individual file (see section “Displaying file attributes” on page 132).

For the catalog entries for a VSAM cluster, the function works as follows:

- If the name of the main entry is specified in the command, the entire cluster is deleted.
- Data and index parts of VSAM clusters cannot be deleted individually.

3.7.7 Deleting directories

Users of openFT partner systems can delete z/OS directories, i.e.

– PO data sets

– PDSE data sets

– openEdition directories

To do this, the FTDELDIR command can be used in z/OS.

It is also possible to delete PO and PDSE data sets if they are not empty.

Note that under openFT for z/OS up to Version 9.0 it was possible to delete PO and PDSE data sets as “files” from a partner system, for instance using the command ftdel (Unix/Windows systems) or DELETE-REMOTE-FILE (BS2000/OSD). In openFT V10 for z/OS (or higher), the command ftdeldir/FTDELDIR or DELETE-REMOTE-DIRECTORY must always be used.

4 Menu interface for the FT user

This chapter is intended for openFT users who use the Dialog Monitor to transfer ISPF files with openFT under TSO and who may possibly want to use openFT-AC to edit the file transfer security settings for their user IDs. The earlier distinction between one menu interface for administrators and another for users has been abolished. Both now see the same entry menu (Primary Option Menu). The FT administrator naturally has more rights.

4.1 Creating an openFT instance

16 so-called openFT instances may be present in parallel on a z/OS system. In themselves, these instances represent complete openFT systems each with their own request files and partner lists, their own addresses and, in some cases, their own FTAC settings. Instances are set up by the FT administrator.

In order to use the menu interface, you must concatenate the corresponding libraries:

– the CLIST OPENFT.PANEL.CLIST
– the panel library OPENFT.PANELS
– the message library OPENFT.PANEL.MSG

If an instance other than STD is to be used or if an SVC number other than 211 is used for the openFT subsystem, you must also allocate the CONN file of the required instance via the DD name OPENFT:

<openft qualifier>.<inst>.CONN

The FT administrator defines the specifications for the OPENFT QUALIFIER and the name of the instance (<inst>) when setting up the instance. For information on where to find these files on your z/OS system an whether an SVC number other than 211 is used, please consult your system administrator.

Even if the STD instance with SVC number 211 is used, it is urgently recommended that the file <openft qualifier>.<inst>.CONN is allocated.

Even if the STD instance with SVC number 211 is used, it is urgently recommended that the file <openft qualifier>.<inst>.CONN is allocated.If the openFT commands (NCOPY, NSTATUS, ... etc.) are to be used in a TSO session, it is still important to declare the NCLOAD in which the commands are located or to add it to the search path. This is done using the command TSOLIB (see the example below).

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

Example of this type of CLIST

```
allocate file(sysproc) dataset('isp.sispclib' -
'OPENFTQU.STD.CLIST'                          -
'OPENFTQU.OPENFT.PANEL.CLIST' ) reuse shr
allocate file(ispplib) dataset('isp.sisppenu' -
'OPENFTQU.OPENFT.PANELS') reuse shr
allocate file(ispmlib) dataset('isp.sispmenu' -
'OPENFTQU.OPENFT.PANEL.MSG') reuse shr
ALLOC DSNAME('OPENFTQU.STD.CONN') DDNAME(OPENFT) SHR REUSE
TSOLIB ACT DATASET('OPENFTQU.OPENFT.NCLOAD')
```

4.2 General

All openFT user functions are supported by the ISPF product provided that this product is installed in the z/OS system. In this case, the FT user is provided with the user-friendly ISPF interface with the usual choice of menus and data entry panels, immediate warnings in the event of errors, help functions etc. When the FT user uses this menu interface, the corresponding commands are issued internally. The panels of the menu interface are described below.

A description of how to enter the commands NCOPY, NCANCEL and NSTATUS directly under TSO is given in the chapter “Command interface” on page 147.

4.3 Software requirements

The commands involved cannot be called via ISPF panels unless IBM Program Product ISPF (Interactive System Productivity Facility), Version 2 or later, is installed in the system.

The System Administrator Guide "openFT for z/OS - Installation and Administration" provides a description of how to install the libraries containing the openFT panel definitions, CLISTs and messages.

If you want to use these panels, there are two options open to you:

a) Normally, you should be able to call the "Primary Option Menu" of this menu interface (see page 144) from the ISPF panel hierarchy of your system. This assumes that the FT administrator has concatenated the libraries named above with those of the installed ISPF system and provided an option for calling the "Primary Option Menu" of this menu interface when installing openFT. Please ask your FT administrator whether this is the case.

b) Otherwise, you must proceed as follows:

   – You must concatenate the libraries with the openFT panel definitions, CLISTs and messages for your TSO session with the system libraries yourself. Details are described in the pervious section.

   – Still in TSO mode, you then call a CLIST which calls the entry panel of the menu interface:

     ```
     EX OPENFT.PANEL.CLIST(FJMENU)
     ```

     This CLIST contains the line:

     ```
     IPSTART CMD(FTSETUP) NEWAPPL(OPFT)
     ```

4.4 Representation and utilization

The structure of the menu system is described below. Detailed online help on each panel is available by pressing the F1 key.

Terminal operation is subject to the rules that usually apply with IBM ISPF:

– The ENTER key causes terminal input to be passed on and, where appropriate, verified. The particular reaction that follows depends on the panel currently displayed.

– In many cases, a data entry panel appears in which you can or must make entries. An action is then executed which has the same effect as issuing the corresponding FT command. In the next step, the message issued by openFT in response to this action is displayed on the screen. You exit this display in the usual manner using END, RETURN or the "jump function" (see below).

– The END command causes a return to the panel preceding the current panel in the panel hierarchy. In this case, no action is usually executed. This enables you to cancel actions which you have selected by mistake.

– The RETURN command causes a return to the Primary Option Menu. In this case, too, no action is executed.

– The "jump function" of ISPF (calling a sequence of panels in one step e.g. "=p.3") is supported. In this case, the effect of the END command differs from the one described above: as usual with ISPF, "the panel preceding the current panel in the panel hierarchy" is interpreted as the panel from which the "jump" was made; so the END command causes that panel to be displayed. In this case, too, no action is executed.

– Function keys PF1 through PF12 (or through PF24) can be used as usual with ISPF.

– If syntactically incorrect input or another input error has been detected by the menu system, a short message is displayed in the top right-hand corner of the screen indicating the error. At the same time, the cursor is positioned at the input field concerned. Subsequent entry of the HELP command causes a more detailed message to be displayed in the third line on the screen. Repeating the HELP command causes a help panel to be displayed.

– The HELP panels for the individual functions form a hierarchy; you can therefore use the usual commands to "browse" through these help panels (e.g. ENTER to display the next help panel, BACK to display the previous help panel, etc.).

– Data you have entered in data entry panels is generally deleted as soon as you exit the panel. Exceptions are noted as appropriate for each panel.

– The data you have entered is not deleted, however, if the same data entry panel is displayed again following execution of the function (ENTER). This is the case for a number of functions which can be effectively repeated a number of times in succession (e.g. the function ADD REMOTE SYSTEM TO NETWORK DESCRIPTION). In this case, the data you have entered is also displayed once again and you can modify it before executing the function again. This applies until you finally exit the panel using END (or RETURN or the "jump function").

– In the case of "string" type input fields, the uppercase/lowercase notation is taken over, otherwise all inputs are converted to uppercase.

– The equals sign "=" has its usual ISPF navigation function (e.g. "=x" to exit the interface). For this reason, it is not possible to pass openFT any values that start with "=" via the interface.

Refer to the relevant IBM manuals for further information about ISPF.

The entries you can or must make in the fields of the data entry panels correspond to the parameter values which you must specify for the corresponding FT administration command. They are described in the chapter entitled chapter “Command interface” on page 147).

The messages issued by openFT in response to your actions are also the same as those issued at the command interface. These messages and their meanings are given in the appendix (see page 381).

openFT displays the "PRIMARY OPTION MENU" illustrated on the next page as the entry panel. Menu items 5 and 6 in this menu are only available if openFT-AC is installed.

PRIMARY OPTION MENU

```
---  openFT - PRIMARY OPTION MENU  -----------------

OPTION  ===>

   1  ADMINISTRATION

   2  FILE TRANSFER REQUESTS

   3  EXECUTE REMOTE COMMANDS

   4  EXECUTE REMOTE FTADM COMMANDS

   5  ADMISSION SETS

   6  ADMISSION PROFILES

   INSTANCE IN USE ===>  STD

   COMMAND DISPLAY ===>  Y  (Y/N)

   ------------------------------------------------------
   | Copyright (C) Fujitsu Technology Solutions, 2012   |
   ------------------------------------------------------

  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
```

This is the first panel of the openFT panel hierarchy when FTAC is used. It is qualified as the "Primary Option Menu", which means that it is the panel to which you return from any subsequent openFT panel after entering the RETURN command.

You enter YES or NO in the COMMAND DISPLAY field in order to specify whether or not the FT commands which correspond to the functions you select in the subsequent menus are to be displayed on the screen, together with all the parameters which correspond to your entries in the data entry panel, if applicable.

Provided you do not change this setting, it remains valid throughout the session and is retained after the session is terminated.

The following list illustrates the hierarchy of the subsequent menus and functions that can be accessed from the Primary Option Menu. FTAC-specific items are present only if openFT-AC is installed.

1 ADMINISTRATION (only for administrators, see the Administrator Guide)

2 FILE TRANSFER REQUEST

  1 ENTER FILE TRANSFER REQUEST

  2 SHOW/MODIFY/CANCEL FILE TRANSFER REQUEST(S)

  3 SHOW LOGGING RECORDS OR FILES

  4 SHOW ALLOWED PARTNER SYSTEMS

3 EXECUTE REMOTE COMMANDS

4 EXECUTE REMOTE FTADM COMMANDS

5 ADMISSION SETS

6 ADMISSION PROFILES

  (Create, list, modify, delete FT admission profiles)

4.5 Error messages

The messages issued by openFT in response to your actions at the menu interface are the same as those issued at the command interface. These messages are explained in the appendix (see page 381).

Errors you make while entering data in the panels are displayed in the usual way in ISPF (output of a short message or, if the HELP command is issued, a long message).

Short messages and long messages can also occur for other reasons, however, e.g. in the event of errors when accessing temporary files. There is the following temporary file:

<inst>.FJCMD.TMP.OUT

When some of the menu interface functions are executed, a temporary PS data set is created to buffer the command. In SYSPLEX mode, the suffix from the variable SYSNAME that is populated by the system is also appended: <inst>.FJCMD.TMP.OUT.<SYSNAME>.

This data set is usually deleted again after the function has been executed.

inst: Instance name of the currently set openFT instance

If it is not possible to create a temporary file, the following messages are output:

Short Message `I/O ERROR`

Long Message: `ERROR OCCURRED ON ACCESSING TEMPORARY OUTPUT FILE`

5 Command interface

This chapter contains a functional description of the openFT commands, as well as detailed descriptions of the individual commands.

The functional command description provides a quick overview of which commands are available for which tasks.

This is followed by an explanation of how to enter the commands and of the notational conventions used in the command descriptions.

Finally, the commands are described in alphabetical order.

5.1 Setting an openFT instance

16 so-called openFT instances may be present in parallel on a z/OS system. In themselves, these instances represent complete *open*-FT systems each with their own request file, partner list, their own addresses and, in some cases, their own FTAC settings. Instances are set up by the FT administrator.

If an instance other than STD is to be used or if an SVC number other than 211 is used for the openFT subsystem, you must also allocate the CONN file of the required instance via the DD name OPENFT:

<openft qualifier>.<inst>.CONN

The FT administrator defines the specifications for the OPENFT QUALIFIER and the name of the instance (<inst>) when setting up the instance. For information on where to find these files on your z/OS system an whether an SVC number other than 211 is used, please consult your system administrator.

Even if the STD instance with SVC number 211 is used, it is urgently recommended that the file <openft qualifier>.<inst>.CONN is allocated.

If the openFT commands (NCOPY, NSTATUS, ... etc.) are to be used in a TSO session, it is still important to declare the NCLOAD in which the commands are located or to add it to the search path. This is done using the command TSOLIB (see the example below).

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

Example of this type of CLIST

```
ALLOCATE FILE(SYSPROC) DATASET('ISP.SISPCLIB' REUSE SHR
ALLOC DSNAME('OPENFTQU.STD.CONN') DDNAME(OPENFT) SHR REUSE
TSOLIB ACT DATASET('OPENFTQU.OPENFT.NCLOAD')
```

5.2 Functional command overview

The following overview shows the FT and FTAC user commands as they relate to individual jobs. The following user groups are distinguished here:

FT user
: Person who uses functions of the product openFT but has no rights as FT administrator.

FT administrator
: Person who manages the product openFT on a computer.

FTAC user
: Person who can manage admission records for his/her own user ID but does not have the rights of an FTAC administrator.

FTAC administrator
: Person who manages the product openFT-AC on a computer.

In a number of commands additional options are available to the FT or FTAC administrator which enable him/her to perform the associated actions system-wide. In addition, there are commands which only the FT or FTAC administrator may call. All administrator-specific commands and command options are described in the manual openFT for z/OS - Installation and Administration.

5.2.1 FT command overview

Showing openFT instances		
Show openFT instance	FTSHWINS	page 258
Transfer files		
Submit asynchronous FT request	NCOPY	page 312
Submit synchronous FT request	FTSCOPY	page 243
Cancel FT requests	NCANCEL, also FTCANREQ	page 308
Show information on FT requests	NSTATUS, also FTSHWREQ	page 353
Modify FT request queue	FTMODREQ	page 240

File management

Show attributes of file/files in a remote system	FTSHW	page 248
Change file attributes in a remote system	FTMOD	page 204
Delete file in a remote system	FTDEL	page 188
Create directory in a remote system	FTCREDIR	page 165
Change attributes of a directory in a remote system	FTMODDIR	page 204
Delete directory in a remote system	FTDELDIR	page 191

Execute remote commands

Execute commands in the remote system	FTEXEC	page 197

Logging Function

Show log records or log files	FTSHWLOG	page 259
Display information on reason codes in the logging records	FTHELP	page 202

Monitoring

Show monitoring data	FTSHWMON	page 276

Obtain information on openFT

Display operating parameters	FTSHWOPT	page 287
Display partner systems	FTSHWPTN	page 298

5.2.2 FTAC commands overview

openFT-AC must be installed in order to use the following commands:

Edit FTAC admission profiles

Create admission profile	FTCREPRF	page 168
Delete admission profile	FTDELPRF	page 194
Modify admission profile	FTMODPRF	page 218
Show admission profile	FTSHWPRF	page 294

Edit FTAC admission sets

Modify admission set	FTMODADS	page 209
Show admission set	FTSHWADS	page 254

Show partner systems

Display partner systems and security levels	FTSHWRGE	page 305

5.3 Entering FT commands

Please remember the following when entering commands:

– You must insert commas to separate the individual operands of a command, e.g.

```
NCOPY TRANSFER-DIRECTION=TO,PARTNER-NAME=ZENTRALE,LOCAL-PARAMETER =...
```

– If quotes appear in a value assignment which is itself enclosed in quotes, they must be entered twice.

– If there is no default value marked (by underscoring) for an operand, then it **must** be specified with a valid value (mandatory operand).

– A distinction is made between positional operands and keyword operands. Positional operands are uniquely determined by their position in the command. Keyword operands are uniquely determined by their keyword, for example `TRANSFER-DIRECTION=...` There are a number of considerations to be borne in mind when specifying such operands (see below).

– You can abbreviate your entries for commands and operands, always ensuring that your entries retain their uniqueness. You can also use positional operands if you wish. Short forms and long forms can be mixed at will. Certain abbreviated forms of keywords and a number of positional operands are guaranteed for openFT. In the command representation the recommended abbreviation is shown in **bold**. This means that you will find these options unchanged in subsequent versions. This means, therefore, that to be “on the safe side”, you should form the habit of entering these commands in their abbreviated form. You should take particular care to use the guaranteed abbreviated forms in procedures, as this will ensure their continued executability in subsequent versions. The recommended abbreviations are used in the examples shown in this chapter. The possible abbreviations are listed for the individual command formats.

– If a structure is preceded by an introductory operand value, then the opening parentheses must immediately follow this operand value. Example: *BS2000 is an introductory operand value in REM=*BS2000(...). Introductory operand values may be omitted if there is no risk of ambiguity.

– The asterisk (*) that precedes constant operand values may be omitted if there is no risk of ambiguity. Please ensure that it is not a guaranteed abbreviation.

– Comments may be included in FT user commands using the form "..."; the normal method of including comments in other TSO commands using the form /*...*/ is not permitted.

When you enter commands, the value assignments for the operands may be specified in positional form, in keyword form or in mixed form.
Please note the following:

– When you perform value assignments in positional form, the first value is assigned to the first operand in the command, the second value to the second operand etc.

– Values assigned in positional form are separated by commas. You must also enter a comma for each operand for which no value is assigned.

– If two values are assigned to an operand, the last value to be assigned always applies. This also applies to parameter specifications in introductory operand values within the corresponding structure brackets. However, for the sake of clarity, double assignments should generally be avoided.

– If you mix the different forms of operand value assignments (positional and keyword form), then you must observe the correct sequence. Note that you can start your input with positional operands and follow these with keyword operands but not the other way round!

– Since there is a possibility that the sequence of operands may change in subsequent versions, only keyword operands should be used in procedures.

Continuation lines in FT commands in z/OS

An NCOPY command may consist of more than one line. When entering an NCOPY command with continuation lines at a TSO terminal, you simply continue writing on the next line on the screen.
If an NCOPY command with continuation lines is issued in a CLIST or REXX procedure or in a batch job as data for the IBM utility IKJEFT01, a hyphen "-" or a plus sign "+" is used as the continuation character. Refer to the IBM manuals for more details.

Differentiation between uppercase and lowercase letters

It may be important to differentiate between uppercase and lowercase letters in the parameters.

openFT handles the letters contained in the command string according to the following rules:

1. If the command string received by openFT contains only uppercase letters,
   – all letters outside the quotation marks remain uppercase;
   – letters enclosed in quotation marks are converted to lowercase.
   – alphanumerically specified FTAC transfer admissions are converted into lowercase letters

2. If any part of the command string received by openFT except the command name (NCOPY) contains a lowercase letter,
   - all letters outside the quotation marks are converted to uppercase;
   - alphanumerically specified FTAC transfer admissions are converted into lowercase letters
   - letters enclosed in quotation marks are not converted. These letters are retained in the form in which they were entered.

This has the following consequences for command input:

If parameter values consisting of uppercase letters (or of both uppercase and lowercase letters) enclosed in quotation marks are to be entered, you must ensure that

- the command contains at least one lowercase letter (at any position except in the command name) and
- openFT receives this command string in the same form (with no conversion).

This means that

- In a CLIST or REXX procedure, you must use the statement CONTROL ASIS (or CONTROL NOCAPS) to ensure that the command string is not converted to uppercase before execution.
- You can also use the menu interface (see page 139); here, the relevant fields are not converted to uppercase (see the description of the input fields in the data entry panels).
- When the TSO command processor is called in a batch job (IBM utility IKJEFT01, see section “Using openFT in z/OS systems without the TSO interactive system” on page 464), letters are not converted to uppercase.

These rules also apply to the hexadecimal digits A through F in entries of the form <x-string m..n> which expect the partner system to be specified in uppercase letters.

5.4 Command syntax representation

The command format consists of a field with the command name. All operands with their legal values are then listed. Operand values which introduce structures and the operands dependent on these operands are listed separately. The syntax of the command representation is explained in the following three tables.

table 1: Notational conventions

The meanings of the special characters and the notation used to describe command and statement formats are explained in table 1.

table 2: Data types

Variable operand values are represented in SDF by data types. Each data type represents a specific set of values. The number of data types is limited to those described in table 2.

The description of the data types is valid for the entire set of commands/statements. Therefore only deviations (if any) from the attributes described here are explained in the relevant operand descriptions.

table 3: Suffixes for data types

Data type suffixes define additional rules for data type input. They contain a length or interval specification.

The description of the data type suffixes is valid for the entire set of commands/statements. Therefore only deviations (if any) from the attributes described here are explained in the relevant operand descriptions.

Metasyntax

Representation	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters denote keywords (command, statement or operand names, keyword values) and constant operand values. Keyword values begin with *	**HELP-SDF** **SCREEN-STEPS** = ***NO**
UPPERCASE LETTERS in boldface	Uppercase letters printed in boldface denote guaranteed or suggested abbreviations of keywords.	**GUID**ANCE**-MODE** = ***Y**ES
=	The equals sign connects an operand name with the associated operand values.	**GUID**ANCE**-MODE** = ***NO**
< >	Angle brackets denote variables whose range of values is described by data types and suffixes (see Tables 2 and 3)).	**SYNTAX-F**ILE = <filename 1..54>
Underscoring	Underscoring denotes the default value of an operand.	**GUID**ANCE**-MODE** = ***NO**
/	A slash serves to separate alternative operand values.	**NEXT-FIELD** = ***NO** / ***Y**ES
(...)	Parentheses denote operand values that initiate a structure.	,**UNGUID**ED**-DIA**LOG = ***Y**ES(...) / ***NO**
[ ]	Square brackets denote operand values which introduce a structure and are optional. The subsequent structure can be specified without the initiating operand value.	**SELECT** = [***BY-ATTR**IBUTES](...)
Indentation	Indentation indicates that the operand is dependent on a higher-ranking operand.	,**GUID**ED**-DIA**LOG = ***Y**ES(...) ***Y**ES(...) **SCREEN-STEPS** = ***NO** / ***Y**ES

Table 1: Metasyntax (part 1 of 2)

Representation	Meaning	Examples
|	A vertical bar identifies related operands within a structure. Its length marks the beginning and end of a structure. A structure may contain further structures. The number of vertical bars preceding an operand corresponds to the depth of the structure.	**SUP**PORT = ***TAPE**(...) ***TAPE(...)** | **VOL**UME = ***ANY**(...) | ***ANY**(...) | | ...
,	A comma precedes further operands at the same structure level.	**GUID**ANCE-**MODE** = ***NO** / ***Y**ES ,**SDF-COM**MANDS = ***NO** / ***Y**ES
list-poss(n):	The entry “list-poss” signifies that a list of operand values can be given at this point. If (n) is present, it means that the list must not have more than n elements. A list of more than one element must be enclosed in parentheses.	list-poss: ***SAM** / ***ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <filename 1..54>
Alias:	The name that follows represents a guaranteed alias (abbreviation) for the command or statement name.	**HELP-SDF** Alias: **HPSDF**

Table 1: Metasyntax (part 2 of 2)

Data types

Data type	Character set	Special rules
alphanum-name	A…Z 0…9 $, #, @	
c-string	EBCDIC character	Must be enclosed within single quotes; the letter C may be prefixed; in the case of file names in z/OS it must be prefixed; any single quotes occurring within the string must be entered twice.
composed-name	A…Z 0…9 $, #, @ Hyphen Period	Alphanumerical string that can be subdivided into multiple substrings by periods or hyphens.
date	0…9 Structure identifier: hyphen	Input format: yyyy-mm-dd yyyy: year; optionally 2 or 4 digits mm: month dd: day Only date specifications between 1.1.2000 and 19.1.2038 are possible. If the year is specified in 2-digit form, 2000 is added to the number
filename	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<filename>' Input format partially qualified: :<prefix>:<filename> :<prefix>: Optional specification of file organization; enclosed in colons; can assume the following values: :S: for PS :O: for PO :E: for PDSE :L: for PO or PDSE :V: for VSAM

Table 2: Data types (part 1 of 3)

Data type	Character set	Special rules
		<first-qual> "first level qualifier" User ID (max. 7 characters, character range A…Z, 0…9, $, #, @; may not begin with a digit) or alias (max. 8 characters) <filename> partially qualified file name; the syntax of z/OS file names depends on the file organization; refer to the overview as of page 57
filename-prefix	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' or ':<prefix>:<first-qual>.<partname>/' Input format partially qualified: :<prefix>:<partname>. or :<prefix>:<partname>/ <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period or a slash.
integer	0…9, +, -	+ or -, if specified, must be the first character.
name	A…Z 0…9 $, #, @	Must not begin with 0...9.
number	0...9 A...F	Message number/return code

Table 2: Data types (part 2 of 3)

Data type	Character set	Special rules
partial-filename	A…Z 0…9 $, #, @ hyphen period	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' Input format partially qualified: :<prefix>:<partname>. <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period.
text	freely selectable	For the input format, see the relevant operand descriptions.
time	0…9 structure identifier: colon	Time-of-day entry: Input format: { hh:mm:ss / hh:mm / hh } hh: hours mm: minutes ss: seconds } Leading zeros may be omitted Valid entries are between 00:00:00 and 23:59:59.
x-string	Hexadecimal: 00…FF	Must be enclosed in single quotes; must be prefixed by the letter X. There may be an odd number of characters.

Table 2: Data types (part 3 of 3)

Suffixes for data types

Suffix	Meaning
x..y	With data type "integer": interval specification
	x minimum value permitted for "integer". x is an (optionally signed) integer.
	y maximum value permitted for "integer". y is an (optionally signed) integer.
x..y	With the other data types: length specification For data types date and time the length specification is not displayed.
	x minimum length for the operand value; x is an integer.
	y maximum length for the operand value; y is an integer.
	x=y the length of the operand value must be precisely x.

Table 3: Suffixes for data types

Meaning of operands

After the format of each command there is a detailed description of all the operands, the possible value assignments and their functions.

Otherwise the same metasyntax is used in describing operands as in the representation of the command formats (see above).

The following characters are regarded as constants in describing the operands: "." (period), "(" (open bracket), ")" (close bracket), " ' " (single quote), "$" (dollar sign), and also the character combinations ":V:", ":L:", ":S:", ":O:" and ":E:" i.e. they must be specified when the command is entered. Where this occurs the syntactical components of the operand value must follow one after another without any gaps.
"±" has the usual meaning "+" or "-".

Example

Possible entries for the local operand FILE are as follows:

```
ABC                 'USER1.ABC',                    (1)
GROUP1.G1234V01     'USER1.GROUP1.G1234V01'         (2)
GROUP2(+27)         'USER1.GROUP2(+27)'             (3)
GROUP3(0)           'USER1.GROUP3(0)'               (4)
:V:VSDAT            ':V:USER1.VSDAT'                (5)
PDS1(DEF)           'USER1.PDS1(DEF)',              (6)
:L:PODS2            ':L:USER1.PODS2'                (7)
./directory5/abcd   /u/user002/directory5/abcd      (8)
```

Key

(1) Name of a PS data set

(2) Name of an absolute generation data set (PS data set) (this has the same syntax as the name of a normal PS data set, with the exception of the last partial name, which must have a special format)

(3) Name of a relative generation data set (PS data set)

(4) Name of a relative generation data set (PS data set), special case "current generation" (may only be a send file)

(5) Name of a VSAM file of the type "entry sequenced"

(6) Name of a PO or PDSE member

(7) Name of an entire PO or PDSE data set

(8) Pathname of an openEdition file (absolute and relative)

More details on the syntax rules for file names, passwords, user IDs and account numbers in openFT can be found in the respective sections in chapter 3.

5.5 Command return codes

The TSO commands supply a return code that provides information about whether command processing has succeeded or failed. It is stored in the TSO's system variable ("control variable") &LASTCC. A return code other than 0 is generated only if a corresponding message is output at the terminal. These messages are described in the Appendix (page 411ff).

This return code may have the following values:

`Return-Code = 0`:

The command was accepted. (Corresponds, for example, to the message FTR0000 or FTR0008 at the terminal.)

`Return-Code = 4`:

The command was accepted with a minor warning, for example if no corresponding administration objects were found.

`Return-Code = 8`:

Reserved

`Return-Code = 12 (or > 12)`:

The command was rejected due to an error. The request was not accepted.

The TSO commands can also be started in response to an ftexec command that was started in a remote Unix or Windows partner system. The partner system is sent either the return code 0 (if the command was accepted) or 12 (if the command was terminated with an error).

5.6 Output in CSV format

The output of some SHOW commands in openFT and openFT-AC can be optionally requested in CSV (**C**haracter **S**eparated **V**alues) format. CSV is a popular format in the PC environment in which tabular data is defined by lines. Output in CSV format is offered for the following commands:

– FTSHW

– FTSHWADS

– FTSHWLOG

– FTSHWMON

– FTSHWOPT

– FTSHWPTN

– FTSHWPRF

– FTSHWRGE

– NSTATUS

Many programs such as spreadsheets, databases, etc., can import data in CSV format. This means that you can use the processing and presentation features of such programs on the CSV outputs of the command listed above.

The field names of the CSV outputs are described in the appendix.

The first line is the header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in a record.** In other words, the order of columns is determined by the order of the field names in the header line.

5.7 FTCREDIR Create remote directory

Note on usage

User group: FT user

Functional description

With the FTCREDIR command, you can create a directory in an FT partner system. In remote z/OS systems you cannot create PO or PDSE dataset with the command FTCREDIR.

Format

FTCREDIR

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>
,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99.

DIRECTORY-NAME =
Name of the directory in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
If the directory name is specified with a mounted Public Volume Set (BS2000/OSD) then the request is rejected with error message FTR2202.

PASSWORD =
If the file system or the parent directory only permits the directory to be created with a password, you can specify this here.
This is only possible in the case of partner systems which support this type of password.

PASSWORD = *NONE
No password is required to create the directory.

PASSWORD =
<integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password giving permission to create the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION =
<alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. The alphanumeric input is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD =
<c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

Example:

The directory `Dir1` is to be created in the remote Unix system under the user ID with the transfer admission `transadm`:

```
ftcredir partux,c'Dir1',,transadm
```

5.8 FTCREPRF Create admission profile

Note on usage

User group: FTAC user and FTAC administrator

A prerequisite for using this command is the use of openFT-AC.

Functional description

All FTAC users can use FTCREPRF to set up their own admission profiles under their user IDs. Users must activate admission profiles predefined by the FTAC administrator with FTMODPRF (see page 218ff) before they can be used. Profiles predefined by the FTAC administrator may be used immediately if the FTAC administrator also possesses the SU privilege.

– It is possible to create an admission profile for "pre-processing" or "post-processing". To do this, the FILE-NAME operand must start with the pipe symbol '|'. After this has been done, one or more TSO commands can be specified. For detailed information refer to the section “Preprocessing and postprocessing” on page 111.

Format

(part 1 of 2)

FTCREPRF

NAME = ***STD** / <alphanum-name 1..8>

,**PASS**WORD = **_*NONE_** / <alphanum-name 1..8>

,**TRANS**FER-**AD**MISSION = ***NOT-SPEC**IFIED / <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)

<alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)

VALID = **_*YES_** / ***NO**

,**USAGE** = **_*PRIVATE_** / ***PUB**LIC

,**EXP**IRATION-**DATE** = **_*NOT-REST_**RICTED / <date 8..10>

,**PRIV**ILEGED = **_*NO_**

,**IGN**ORE-**MAX-LEV**ELS = **_*NO_** / ***YES** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

OUTBOUND-**SEND** = **_*NO_** / ***YES**

,**OUT**BOUND-**REC**EIVE = **_*NO_** / ***YES**

,**IN**BOUND-**SEND** = **_*NO_** / ***YES**

,**IN**BOUND-**REC**EIVE = **_*NO_** / ***YES**

,**IN**BOUND-**PROC**ESSING = **_*NO_** / ***YES**

,**IN**BOUND-**MAN**AGEMENT = **_*NO_** / ***YES**

,**USER-AD**MISSION = **_*OWN_** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = **_*OWN_** / <name 1..8>

,**ACCOUNT** = **_*OWN_** / ***NOT-SPEC**IFIED / ***NONE** / <alphanum-name 1..40> / <c-string 1..40>

,**PASS**WORD = **_*OWN_** / <alphanum-name 1..8> / ***NONE**

,**INIT**IATOR = (**_*LOC_**AL, **_*REM_**OTE) / list-poss(2): ***LOC**AL / ***REM**OTE

,**TRANS**FER-**DIR**ECTION = **_*NOT-REST_**RICTED / ***FROM**-PARTNER / ***TO**-PARTNER

,**PART**NER = **_*NOT-REST_**RICTED / list-poss(50): <text 1..200 with-low>

,**MAX-PART**NER-**LEV**EL = **_*NOT-REST_**RICTED / <integer 0..100>

(part 2 of 2)

,**FILE-NAME** = ***NOT-REST**RICTED / <filename1..59> / <c-string 1..512 with-low> / ***EXP**ANSION(...)

,***EXP**ANSION(...)
| **PREF**IX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>

,**FILE-PASS**WORD = ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

,**PROC**ESSING**-AD**MISSION = ***SAME** / ***NOT-REST**RICTED / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
| **USER-ID**ENTIFICATION = ***SAME** / ***NOT-REST**RICTED / <name 1..8>
| ,**ACC**OUNT = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..40> / <c-string 1..40>
| ,**PASSW**ORD = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

,**SUCC**ESS-PROCESSING = ***NOT-REST**RICTED / ***NONE** / <c-string 1..1000 with-low> / ***EXP**ANSION(...)

***EXP**ANSION(...)
| **PREF**IX = ***NOT-RES**TRICTED / <c-string 1..999 with-low>
| ,**SUFF**IX = ***NOT-RES**TRICTED / <c-string 1..999 with-low>

,**FAIL**URE-PROCESSING = ***NOT-REST**RICTED / ***NONE** / <c-string 1..1000 with-low> / ***EXP**ANSION(...)

***EXP**ANSION(...)
| **PREF**IX = ***NOT-RES**TRICTED / <c-string 1..999 with-low>
| ,**SUFF**IX = ***NOT-RES**TRICTED / <c-string 1..999 with-low>

,**WRITE**-MODE = ***NOT-REST**RICTED / ***NEW**-FILE / ***REP**LACE-FILE / ***EXT**END-FILE

,**FT-FUNC**TION = ***NOT-REST**RICTED / list-poss(5): ***TRANS**FER**-F**ILE / ***MOD**IFY**-F**ILE**-ATTR**IBUTES / ***READ-DIR**ECTORY / ***FILE**-**PROC**ESSING

,**USER-INF**ORMATION = ***NONE** / <c-string 1..100 with-low>

,**DATA-ENC**RYPTION = ***NOT-REST**RICTED / ***NO** / ***YES**

Operands

NAME = <alphanum-name 1..8>
With NAME, the admission profile is given a name. This name must be unique among all admission profiles on this user ID. If an admission profile with this name already exists, FTAC rejects the command with the message:

```
FTC0100 COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 294ff) can be used to view the already existing names. To obtain this information, the command FTSHWPRF can be entered without operands.

NAME = *STD
Creates a default admission profile for the user ID. You must specify *NOT-SPECIFIED as the transfer admission, because a default admission profile in a request is addressed using the user ID and password. You must not specify the parameters VALID, USAGE and EXPIRATION-DATE for a default admission profile.

PASSWORD =
FTAC password which authorizes you to issue FTAC commands on your user ID, if such a password was defined in your admission set.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

TRANSFER-ADMISSION =
With TRANSFER-ADMISSION, you define transfer admission. If this transfer admission is entered in an FT request instead of the LOGON admission, then the access rights are valid which are defined in this admission profile. This transfer admission must be unique in the entire openFT instance, so that there is no conflict with other transfer admissions which other FTAC users have defined for other access rights. When the transfer admission which you have selected has already been used, then FTAC rejects the command with the message:

```
FTC0101 COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

TRANSFER-ADMISSION = *NOT-SPECIFIED
This entry is used to set up a profile without transfer admission. If the profile is not a default admission profile, it is locked until you specify a valid transfer admission .

TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
The character string must be entered as the transfer admission in the transfer request. The alphanumeric entry is always stored in lower-case letters.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. With this entry, users can be denied access to the profile.

USAGE = *PRIVATE
Access to your profile is denied for security reasons, when someone with another user ID attempts a second time to specify the TRANSFER ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for uniqueness reasons.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2012-03-31 or 12-03-31 for March 31, 2012. The use of the transfer admission is only possible until the given date.

PRIVILEGED =
The FTAC administrator can privilege the profile. FT requests which are processed with a privileged admission profile are not subject to the restrictions which are set for MAX-ADM-LEVEL (see page 256) in the admission set.

PRIVILEGED = *NO
The admission profile is not privileged. As FTAC user you can omit this parameter, because you only can specify *NO.

IGNORE-MAX-LEVELS =
You can determine for which of the six basic functions the restrictions of the admission set should be ignored. The user’s MAX-USER-LEVELS can be exceeded in this way. The MAX-ADM-LEVELS in the admission set can only be effectively exceeded with an admission profile which has been designated as privileged by the FTAC administrator. The FTAC user can set up an admission profile for himself/herself for special tasks (e.g. sending a certain file to a partner system with which he/she normally is not allowed to conduct a file transfer), which allows him/her to exceed the admission set. This profile must be explicitly given privileged status by the FTAC administrator.
If you enter IGNORE-MAX-LEVELS=*YES, the settings for **all** the basic functions are ignored. If you wish to ignore the admission set for **specific** basic functions, you need to do this with the operands explained later in the text.

The following table shows which partial components of the file management can be used under which conditions:

Inbound file management function	**Setting in admission set/extension in profile**
Show file attributes	Inbound sending (IBS) permitted
Modify file attributes	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Rename files	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Delete files	Inbound receiving (IBR) permitted **and** write rule = overwrite in profile
Show directories	Inbound file management (IBF) permitted **and** direction = to partner in profile
Create, rename, delete directories	Inbound file management (IBF) permitted **and** direction = from partner in profile

IGNORE-MAX-LEVELS = *NO
FT requests which are processed with the admission profile are subject to the restrictions of the admission set.

IGNORE-MAX-LEVELS = *YES
*YES allows you to communicate with partner systems whose security level exceeds the specifications of the admission set. Unless you have a privileged profile, you can only exceed the MAX-USER-LEVELS and not the MAX-ADM-LEVELS in the admission set. You must respect the restrictions defined in the admission set by the FTAC administrator. The SHOW-FT-ADMISSION-SET command provides information on the entries made by the FTAC administrator (see example on page 256). This includes information about the current MAX-USER-LEVELS and MAX-ADM-LEVELS settings.

IGNORE-MAX-LEVELS = *PARAMETERS(...)
The following operands can be used to selectively deactivate the default settings for the individual basic functions.

OUTBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “outbound send” is determined by the admission set.

OUTBOUND-SEND = *YES
For the basic function “outbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

OUTBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “outbound receive” is determined by the admission set.

OUTBOUND-RECEIVE = *YES
For the basic function “outbound receive”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “inbound send” is determined by the admission set.

INBOUND-SEND = *YES
For the basic function “inbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial component “display file attributes” of the basic function “inbound file management”.

INBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “inbound receive” is determined by the admission set.

INBOUND-RECEIVE = *YES
You can disregard your settings for “inbound receive” in the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial components of the basic function “inbound file management”:
– delete files, as long as the file attributes are set accordingly,
– modify file attributes, if the basic function “inbound file management” was admitted in the admission set or in the admission profile.

INBOUND-PROCESSING = *NO
The maximum security level which can be reached with the basic function “inbound follow-up processing” is determined by the admission set.

INBOUND-PROCESSING = *YES
For the basic function “inbound follow-up processing”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-MANAGEMENT = *NO
The maximum security level which can be reached with the basic function “inbound file management” is determined by the admission set.

INBOUND-MANAGEMENT = *YES
For the basic function “inbound file management”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The partial component “modify file attributes” of the basic function “inbound file management” only functions if the basic function “inbound receive” was admitted in the admission set or admission profile.

USER-ADMISSION =
USER-ADMISSION specifies the user ID under which the profile is saved. FT requests which work with this admission profile access the given user ID in the local system.
As FTAC user you can specify only your own user ID here.

Please observe the note on PASSWORD=*OWN on page 176.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications are taken from the current LOGON authorization. A possible z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile. This specification consequently generates a profile in the current user ID.

USER-ADMISSION = *PARAMETERS(...)
Specifies the individual components of the user ID.
This allows you to keep FT requests which use this admission profile under an account number other than the current one, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if the current LOGON password corresponds to the preset password.

USER-IDENTIFICATION =
User ID in z/OS.

USER-IDENTIFICATION = *OWN
The user ID is taken from the current LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID to which the profile should belong. As an FTAC user you can only specify your own user ID; the specification corresponds to *OWN.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = *OWN
The account number is taken from the current LOGON authorization.

ACCOUNT = *NOT-SPECIFIED
No account number is defined.
For further details, see “Default account number” on page 126.

ACCOUNT = *NONE
Has the same effect as ACCOUNT = *NOT-SPECIFIED.

ACCOUNT = <alphanum-name 1..8>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which belongs to the user ID specified in the USER-IDENTIFICATION.

You can also specify accounting information which contains the account number to be used.

PASSWORD =
z/OS password which an FT request should use when it works with this admission profile.

PASSWORD = *OWN
When an FT request refers to this admission profile, FTAC uses the BS2000 password valid for at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.

Admission profiles in which PASSWORD is set to its default value via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, the password must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

PASSWORD = *NONE
No password is required for the user ID specified in the USER-IDENTIFICATION.

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the password specified is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile. The transfer direction is always determined from the system in which the admission profile was defined.

TRANSFER-DIRECTION = *NOT-RESTRICTED
With this admission profile, files can be transferred to and from a partner system.

TRANSFER-DIRECTION = *FROM-PARTNER
With this admission profile, files can only be transferred from a partner system to your system. It is not possible to display file attributes/directories (partial components of “inbound file management”).

TRANSFER-DIRECTION = *TO-PARTNER
With this admission profile, files can only be transferred from your system to a partner system. It is not possible to modify file attributes or delete files (partial components of “inbound file management”).

PARTNER =
Specifies that this admission profile is to be used only for FT requests which are processed by a a certain partner system.

PARTNER = *NOT-RESTRICTED
The range of use for this admission profile is not restricted to FT requests with certain partner systems.

PARTNER = list-poss(50): <text 1..200 with-low>
The admission profile only permits those FT requests which are processed with the specified partner systems. A maximum of 50 partner names can be specified. The total length of all the partners may not exceed 1000 characters. You may specify the name from the partner list or the address of the partner system, see also section “Defining the partner computer” on page 99. It is recommended, to use the name from the partner list. The format shown in the long form of the logging output provides an indication of how a partner address should be entered in an FTAC profile.

MAX-PARTNER-LEVEL =
A maximum security level can be specified. The admission profile will then only permit those FT requests which are processed with partner systems which have this security level or lower.
MAX-PARTNER-LEVEL works in conjunction with the admission set. When non-privileged admission profiles are used, the access check is executed on the basis of the smallest specified value.

MAX-PARTNER-LEVEL = *NOT-RESTRICTED
If FT requests are processed with this admission profile, then the highest accessible security level is determined by the admission set.

MAX-PARTNER-LEVEL = <integer 0..100>
All partner systems which have this security level or lower can be communicated with.

When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the moment). No FT requests can be processed with this admission profile.

FILE-NAME =
Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *NOT-RESTRICTED
Permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Only the specified file may be accessed. However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here (see section “Unique file names for receive files” on page 52). When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX (see page 116). You can also directly specify file transfer with file pre- or post-processing here by entering a pipe symbol ’|’ followed by TSO commands.

FILE-NAME = *EXPANSION(PREFIX = <filename 1..58> / <partial-filename 2..50> / <c-string 1..511 with-low>)
Restricts access to a number of files which all begin with the same prefix. If a *filename* is entered in an FT request which works with this admission profile, FTAC sets the *prefix* defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example
– PREFIX=JACK.; an FT request in which FILE-NAME=BOERSE is specified, then accesses the file JACK.BOERSE.
– PREFIX=TOOLS.CLIST/; an FT request in which FILE-NAME=MEMBER01 is specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or post-processing, you should indicate this by entering the pipe symbol ‘|’ at the start of the prefix. The created FTAC profile can then be used only for file transfer with pre- or post-processing since the file name that is generated also starts with a ‘|’. The variable %TEMPFILE can also be used in the filename prefix. You can find detailed information on preprocessing and postprocessing in section “Preprocessing and postprocessing” on page 111.

The maximum length of the entire pre- or post-processing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example
```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

– A file name or file name prefix that begins with the string '|ftexecsv' must be specified for admission profiles that are to be exclusively used for the ftexec command (see “Example 3” on page 186).

– Specify the file name prefix '|*ftmonitor' for admission profiles that are exclusively used for monitoring. A profile of this sort can then be used in the openFT Monitor or in an ft or ncopy command from a Windows or Unix system (see “Example 2” on page 186).

FILE-PASSWORD =
You can enter a password for files into the admission profile. The FTAC functionality then only permits access to files which are protected with this password and to unprotected files. When a FILE-PASSWORD is specified in an admission profile, the password may no longer be specified in an FT request which uses this admission profile. This allows you to permit access to certain files to users in remote systems, without having to give away the file passwords.

FILE-PASSWORD = *NOT-RESTRICTED
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *SAME
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION=*NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows you to keep FT requests which use this admission profile under a different account number, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
Identifies the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = *SAME
The USER-IDENTIFICATION is taken from the USER-ADMISSION.

USER-IDENTIFICATION = *NOT-RESTRICTED
The admission profile does not restrict the user ID for the follow-up processing.

USER-IDENTIFICATION = <name 1..8>
FT requests which are processed with this admission profile are only permitted follow-up processing under this user ID. If another user ID is entered here, the parameter PASSWORD must also be entered. PASSWORD=*SAME is then not valid.

ACCOUNT =
Account number for the follow-up processing.

ACCOUNT = *SAME
The account number is taken from the USER-ADMISSION.

ACCOUNT = *NOT-RESTRICTED
Account number in FT requests which work with the admission profile. The admission profile does not restrict the account with regard to follow-up processing.

ACCOUNT = *NONE
The account number is used which is defined as the default account number of the user ID specified in the USER-IDENTIFICATION at the time the admision profile is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You an also specify accounting information containing the account number to be used.

PASSWORD =
You specify, where applicable, the z/OS password for the user ID specified in the USER-IDENTIFICATION under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn't have such a password (yet).

PASSWORD = *SAME
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
Specifies the password in FT requests which work with the admission profile. The admission profile does not restrict the password with regard to follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use this admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after successful data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *NOT-RESTRICTED
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Note that blanks at the end of the specification are removed in the FT request, when the follow-up command is assembled. Therefore blanks that are needed here, must be included at the beginning of the specification for SUFFIX.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC='"FILE TRANSFER OK"' is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after failed data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (' or ") within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *NOT-RESTRICTED
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE specification which is valid for this FT request. WRITE-MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = *NOT-RESTRICTED
In an FT request which accesses this admission profile, the operand WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE.

FT-FUNCTION =
Permits the restriction of the profile validity to certain FT functions (=file transfer and file management functions), see also page 40.

FT-FUNCTION = *NOT-RESTRICTED
The full scope of FT functions is available. For reasons of compatibility, the specification NOT-RESTRICTED means that FILE-PROCESSING is not premeditated! All other functions are permitted if this value is specified.

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY,*FILE-PROCESSING)
The following file transfer functions are available:

***TRANSFER-FILE**
The admission profile may be used for the file transfer functions “transfer files”, “view file attributes” and “delete files”.

***MODIFY-FILE-ATTRIBUTES**
The admission profile may be used for the file transfer functions “view file attributes” and “modify file attributes”.

***READ-DIRECTORY**
The admission profile may be used for the file transfer functions “view directories” and “view file attributes”.

***FILE-PROCESSING**
The admission profile may be used for the “pre-processing” and “post-processing” file transfer function. The “transfer files” function must also be permitted.

The *FILE-PROCESSING specification is of relevance only for FTAC profiles without a filename prefix. Otherwise the first character of the filename prefix determines whether only normal data transfer (no pipe symbol |) or only pre-processing and post-processing (pipe symbol |) are to be possible with this FTAC profile.

USER-INFORMATION =
Here, you enter a text in the admission profile. This text is displayed with the command FTSHWPRF.

USER-INFORMATION = *NONE
No text is stored in the profile.

USER-INFORMATION = <c-string 1..100 with-low>
Here, you enter a character string containing user information.

DATA-ENCRYPTION =
Restricts the encryption option for user data.

DATA-ENCRYPTION = *NOT-RESTRICTED
The encryption option for user data is not restricted. Both encrypted and unencrypted file transfers are accepted.

DATA-ENCRYPTION = *NO
Only those file transfers which do not have encrypted user data are accepted, i.e. encrypted requests are rejected.
If the request is made in a BS2000 or z/OS, for example, it must be specified there in the NCOPY request DATA-ENCRYPTION=*NO.

DATA-ENCRYPTION = *YES
Only those file transfer requests that have encrypted user data are accepted, i.e. unencrypted requests are rejected.
If the request is made in a BS2000 or z/OS, for example, it must be specified there in the NCOPY request DATA-ENCRYPTION=*YES.

When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP('PR DSNAME(XYZ.')')

Example 1

Jack John wishes to create an admission profile for the following purpose:

Dylan Dack, employee at the Dack Goldmine, has his own z/OS computer. He has to transfer monthly reports on a regular basis to his boss Jack's computer, JACKJOHN, using File Transfer. The file needs to have the name MONTHLY.REPORT.GOLDMINE and is to be printed out after transfer.

Since Jack's admission set does not permit any "inbound" requests, he needs to give the profile privileged status (he/she is permitted to do this, since he is an FTAC administrator). The Goldmine computer has the security level 50. The command required to create such an admission profile is as follows:

```
FTCREPRF NAME=GOLDMOBE,                                    -
         TRANSFER-ADMISSION=MONTHLYREPORTFORTHEBOSS,       -
         IGNORE-MAX-LEVELS=*YES,                           -
         USER-ADM=(STEFAN,XXXX,PASSWD),                    -
         TRANSFER-DIRECTION=*FROM-PARTNER,                 -
         PARTNER=GOLDMINE,                                 -
         FILE-NAME=MONTHLY.REPORT.GOLDMINE,                -
         SUCCESS-PROCESSING=                               -
         'ALLOC DSNAME(PRINT(MONTHLY.REPORT.GOLDMINE))',-
         FAILURE-PROCESSING=*NONE,                         -
         WRITE-MODE=*REPLACE-FILE
```

The short form of this command is:

```
FTCREPRF␣GOLDMOBE,TRANS-AD=MONATSBERICHTFUERDENCHEF,          -
IGN-MAX-LEV=*YES,USER-ADM=(STEFAN,XXXX,PASSWD),               -
TRANS-DIR=*FROM,PART=GOLDMINE,                                -
FILE-NAME=MONATS.BERICHT.GOLDMINE,                            -
SUCC='ALLOC DSNAME(PRINT(MONATS.BERICHT.GOLDMINE))',FAIL=*NONE, -
WRITE=*REPL
```

File management can also be performed with this admission profile (see the specifications for the IGNORE-MAX-LEVELS operand).

Dylan Dack, who keeps the monthly report for the goldmine in his z/OS computer in the file NOTHING.BUT.LIES, can use the following openFT command to send it to the central computer JACKJOHN and print it out there:

```
/NCOPY␣TO,JACKJOHN,(NOTHING.BUT.LIES), -
       REM=*MSP(FILE=*NOT-SPECIFIED,TRANS-AD=MONTHLYREPORTFORTHEBOSS)
```

Example 2

A profile is to be created that only allows monitoring.

```
FTCREPRF MONITOR,,'ONLYFTMONITOR'  -
   ,FILE-NAME=*EXP('|*FTMONITOR ') -
   ,FT-FUN=(*TRANS-F,*FILE-PROC)
```

The openFT Monitor can be started from a Unix or Windows system using this profile with the following command:

```
ftmonitor "-po=10" FTZOS ONLYFTMONITOR
```

Alternatively, the monitoring values can be output as rows to a file (in this case ftzos_data), for instance with the following command:

```
ncopy FTZOS!"-po=10" ftzos_data ONLYFTMONITOR
```

Example 3

If you only want to use FTAC profiles for the ftexec command then you must specify a filename prefix that starts with the character string ’|ftexecsv’.

If a command or command prefix is also to be defined, you must specify it in the following form:

```
FILE-NAME=*EXP('|ftexecsv -p=command-prefix')
```

If the command string or the command prefix set in the profile for calling ftexec contains spaces, it must be enclosed in double quotes ("). Any double quotes in the command string must be entered twice.

If the entire command string is specified as a file name in the profile for ftexec, you can only specify a space (' ') as the command name when calling ftexec. The FTAC profile does not prevent a caller of ftexec from specifying further command parameters.

Example 4

You want to create a profile which can be used to run precisely one file processing command. A number of logging records are output in the example below.

```
FTCREPRF NUR1VORV,,'GetLoggingRecords'            -
,USER-ADMISSION=(STEFAN,xxxx,password)            -
,FILE-NAME=*EXP('|ftexecsv -p="FTSHWLOG ,"')      -
,FT-FUN=(*TRANS-F,*FILE-PROC)
```

The following command, for example, can be used to access the profile from a remote system:

– Unix system or Windows system:

```
ftexec FTZOS 3 GetLoggingRecords
```

– BS2000 system:

```
/EXE-REM-CMD FTZOS,'3','GetLoggingRecords'
```

– z/OS system:

```
FTEXEC FTZOS,'3','GetLoggingRecords'
```

5.9 FTDEL
Delete remote files

Note on usage

User group: FT user

Functional description

The FTDEL command can be used to delete a file in an FT partner system. You cannot delete directories with this command including PO and PDSE data sets.

Format

FTDEL

PARTNER = <text 1..200 with-low>

,**FILE** = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***PAR**AMETERS(...)

***PAR**AMETERS(...)
| **USER-ID**ENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
| ,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>
| ,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99

FILE =
The name of the file in the remote FT partner system.

FILE = *NOT-SPECIFIED
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
The name of the file in the remote system. The file name must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

PASSWORD =
The password that provides access to the file in the remote system. If the file in the remote system is password-protected, the password required for deleting files in the remote system must be specified in these operands.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
The password that provides access to the file in the remote system. The password must be specified in the syntax of the remote system and conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Contains specifications on transfer admission to the remote system for file management requests.

TRANSFER-ADMISSION = *NONE
The remote system does not require or recognize user authorization.

TRANSFER-ADMISSION =
<alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an admission profile if the FTAC functionality is in use in the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified here. Uppercase alphanumeric input is converted internally to lowercase.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the ID, the account number, and the password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
User ID in the remote system. The ID must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number for the user in the remote system. The account number must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

PASSWORD =
The password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
The password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and be recognized by the remote system.

Example

From your z/OS system, you want to delete the file FILE which is stored in the partner system HUGO. FTAC is implemented in the remote system. The transfer admission DELETE-ACCESS must be specified to delete the file.

```
FTDEL PARTNER=HUGO,FILE=FILE,TRANSFER-ADMISSION=DELETE-ACCESS
```

Short form:

```
FTDEL HUGO,FILE,,DELETE-ACCESS
```

5.10 FTDELDIR Delete remote directory

Note on usage

User group: FT user

Functional description

With the FTDELDIR command, you can delete a directory in an FT partner system., i.e.:

– a PO or PDSE dataset or an openEdition directory in z/OS

– a PLAM library in BS2000/OSD

– any directory in an Unix system or Windows system

A PO or PDSE dataset also can be deleted, if it is not empty.

Format

FTDELDIR

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>

,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>

,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19> /

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99.

DIRECTORY-NAME =
Name of the file in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
If the directory name is specified with a mounted Public Volume Set (BS2000/OSD) then the request is rejected with error message FTR2202.

PASSWORD =
Password making it possible to access the directory in the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password allowing the user to delete the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system required to execute the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. In the case of alphanumeric input, uppercase is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

Example:

Delete the empty directory `Dir1` on the Unix system `partux` under the transfer admission `transadm`.

```
DELETE-REMOTE-DIRftdeldir partux,c'Dir1',,transadm
```

5.11 FTDELPRF Delete admission profile

Note on usage

User group: FTAC user and FTAC administrator

A prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTDELPRF, you can delete all admission profiles of which you are the owner. You should occasionally thin out the set of profiles to ensure that there are no out-of-date admission profiles in your system that could potentially threaten the security of your system.

With SHOW-FT-PROFILE (see page 294ff), you can view the profiles and decide which ones you no longer need.

Format

FTDELPRF

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,**PASS**WORD = ***NONE** / <alphanum-name 1..8>

,**SEL**ECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

TRANSFER-**AD**MISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> /
<c-string 8..32 with-low> / <x-string 15..64>

,**OWN**ER-**ID**ENTIFICATION = ***OWN** / <name 1..8>

Operands

NAME =
You can access the admission profile to be deleted using its name.

NAME = *ALL
Deletes all admission profiles. The FTAC user can delete all of his/her admission profiles with this operand if he/she does not select a special profile with SELECT-PARAMETER.

NAME = <alphanum-name 1..8>
Deletes the admission profile with the specified name.

NAME = *STD
Deletes the default admission profile for your own user ID.

PASSWORD =
You enter the FTAC password which permits you to use FTAC commands with your user ID.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
Specifies the corresponding FTAC password.

SELECT-PARAMETER =
You can enter selection criteria for the admission profiles to be deleted.
FTAC users can address the admission profiles to be deleted using their TRANSFER ADMSSION.

SELECT-PARAMETER = *OWN
Deletes your own admission profiles.

SELECT-PARAMETER = *PARAMETERS(...)
With this structure, you can enter individual selection criteria.

TRANSFER-ADMISSION =
You can use the transfer admission of an admission profile as a selection criterion for deletion.

TRANSFER-ADMISSION = *ALL
Deletes admission profiles irrespective of the TRANSFER-ADMISSION.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Deletes admission profiles for which no transfer admission is specified.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Deletes the admission profile which is accessed with this transfer admission. The alphanumeric entry is always saved in lower-case letters. The FTAC user can only enter the transfer admissions of his/her own admission profiles.

OWNER-IDENTIFICATION =
Deletes a specific owner's admission profile. The FTAC user can only delete his/her own profiles.

OWNER-IDENTIFICATION = *OWN
Deletes your own admission profile.

OWNER-IDENTIFICATION = <alphanum-name 1..8>
The FTAC user can only specify his/her own user ID; the effect corresponds to *OWN.

Example:

A user wants to delete his/her admission profile with the name *Patent*:

```
FTDELPRF PATENT
```

5.12 FTEXEC
Execute remote command

Note on usage

User group: FT user

Functional description

With the FTEXEC command, you can execute operating system commands in the remote system. In the local system, the resulting standard and standard error output can be sent to *STDERR, *STDOUT or to a file.

FTEXEC is only available for openFT partners from Fujitsu Technology Solutions.

On success, FTEXEC returns 0 and if an error occurs it returns 12. The result of the command executed is also transferred.

In the case of output to *FILE, it is possible to specify character sets.

In the case of output to *STDOUT, the character set specified in the local z/OS is used.

Format

FTEXEC

PARTNER = <text 1..200 with-low>

,**CMD**= ***NOT-SPECIFIED** / <c-string 1..400 with-low> (...)
  CODED-CHARACTER-**SET** = ***STD** / <alphanum-name 1..8>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
  ***PAR**AMETERS(...)

***PAR**AMETERS(...)
  USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
  ,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>
  ,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>

,**OUT**PUT = ***STDERR** / ***STDOUT** / ***FILE**(...)

***FILE**(...)
  FILE-NAME = <filename 1..59>
  ,**CODED-CHAR**ACTER-**SET** = ***STD** / <alphanum-name 1..8>

,**DATA**-TYPE = ***CHAR**ACTER / ***BIN**ARY

,**DATA-ENC**RYPTION = ***NO** / ***YES**

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99.

CMD =
Command in the syntax of the remote FT partner system. A command sequence in the remote system can only be processed if the remote system is using an FT product that supports this function.

CMD = *NOT-SPECIFIED
No command string is passed. *NOT-SPECIFIED must be used if an admission profile is specified in TRANSFER-ADMISSION for which a command sequence has been preset.

CMD = <c-string 1..400 with-low>
Command sequence. This command sequence may be a maximum of 400 characters in length, with special characters being counted double (as two characters).

CODED-CHARACTER-SET =
Coding (character set) to be used when reading the data from the standard output of the remote command.

CODED-CHARACTER-SET = *STD
The character set defined as standard in the remote system is used.

CODED-CHARACTER-SET = <alphanum-name 1..8>
The specified character set (CCS) is used. This must be known in the remote system. This specification must not be combined with DATA-TYPE=*BIN.

TRANSFER-ADMISSION =
Contains specifications about the transfer admission in the remote system.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. In the case of alphanumeric input, uppercase is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

OUTPUT =
Specifies where the data generated by the command should be output following transfer in the local system.

OUTPUT = *STDERR
The data is written to *STDERR.

OUTPUT = *STDOUT
The data is written to *STDOUT.

OUTPUT = *FILE(...)
The data is written to a file. Please note that only the data which the command specified with CMD outputs to *SYSLST (BS2000) or *STDOUT (on z/OS) or stdout (on a Unix/Windows system) is written to file.

FILE-NAME = <filename 1..59>
Name of the output file.

CODED-CHARACTER-SET =
Coding (character set) that is to be used to write the data.

CODED-CHARACTER-SET = *STD
The character set predefined by the system is used.

CODED-CHARACTER-SET = <alphanum-name 1..8>
Name of the character set (CCS) that is to be used. This character set must be known in the local system.

This specification must not be combined with DATA-TYPE=*BIN.

DATA-TYPE =
Transfer format for the data.

DATA-TYPE = *CHARACTER
The data is transferred as a text file.

DATA-TYPE = *BINARY
The data is transferred in binary form.

DATA-ENCRYPTION =
Specifies whether the data is to be transferred in encrypted form. The encryption of the request description data is not affected by this operand.

DATA-ENCRYPTION = *NO
The data is transferred unencrypted.

DATA-ENCRYPTION = *YES
The data is transferred encrypted.

Examples

1. The partner is a BS2000 system, output to the local file *ex.out*:

```
FTEXEC BS2PART,'/SH-FT-LOG ,3 ,OUTPUT=SYSLST',(userId,acct,'passw'),
OUTPUT=*FILE(ex.out),DATA-TYPE=*CHAR
```

2. The partner is a Unix system, output to *STDOUT:

```
FTEXEC PARTUX,'ftshwl -nb=10',uxtransadm,,*CHAR
```

3. The partner is a z/OS system:

```
a) FTEXEC ZOS1,'ftshwopt',transadm
```

```
b) FTEXEC ZOS2,'ftshwlog,10,out=*stdout',transadm,out=*file(ex.out)
```

that only the data which the command specified with CMD outputs to *SYSLST (BS2000) or *STDOUT (on z/OS) or stdout (on a Unix/Windows system) is written to file.

5.13 FTHELP
Display information on reason codes in the logging records

Note on usage

User group: FT user and FT administrator

Functional description

You can have the meaning of the reason codes contained in the logging records displayed by the command FTHELP (RC in the output of the command FTSHWLOG in logging records).

Format

FTHELP
<number 1..ffff>

Description

<number 1..ffff>
Stands for a four-digit reason code as it appears in the logging record. Leading zeros can be omitted during input. In an FTAC logging record, the reason code 0000 means that an FTAC admission check has permitted the request. Any other reason code indicates the reason for rejection by FTAC.
The reason code 0000 in an FT logging record indicates that file transfer has terminated successfully. All reason codes other than 0000 indicate failure.

Example

A transfer code is rejected by the local system with the following error message:

```
FTR2046 OPENFT: Local transfer admission invalid.
```

The FTAC administrator uses the command FTSHWLOG (see page 259) to display the relevant FTAC logging record. This is what the output he/she receives looks like:

```
TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
2012-04-24
C       77 15:19:06 3003 >JUMBO    USER001        USER001  ABC
```

The meaning of reason code 3003 can now be determined with the command FTHELP:

```
FTHELP 3003
 3003: Request rejected. Invalid password
```

Thus, the request was rejected because an invalid password was specified.

5.14 FTMOD
Modify remote file attributes

Note on usage

User group: FT user

Functional description

The FTMOD command is used to modify the attributes of a file in an FT partner system. This command does not allow directories (also no PO and PDSE datasets) to be modified.

Depending on the partner involved, the following file attributes can be modified:

openFT partners:

– File name
– Access rights

FTP partners:

– File name

Format

FTMOD

PARTNER = <text 1..200 with-low>

,**FILE** = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***PAR**AMETERS(...)

***PAR**AMETERS(...)
- **USER-ID**ENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
- ,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>
- ,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> /<alphanum-name 1..19>

,**NEW-NAME** = ***SAME** / <filename 1..54> / <c-string 1..512 with-low>

,**FILE-AVAIL**ABILITY = ***UNCHA**NGED

,**STOR**AGE-**ACC**OUNT = ***UNCHA**NGED

,**FUTURE-FILE**-SIZE = ***UNCHA**NGED

,**ACCESS-MODE** = ***UNCHA**NGED / ***READ-ONLY** / ***READ-WRITE** / ***REPL**ACE-**ALL-**BY(...)

***REPL**ACE-**ALL-**BY(...)
- **READ**-FILE = ***NO** / ***Y**ES
- ,**INS**ERT-**DATA**-UNIT = ***NO** / ***Y**ES
- ,**REPLACE**-FILE = ***NO** / ***Y**ES
- ,**EXT**END-FILE = ***NO** / ***Y**ES
- ,**ERA**SE-**DATA**-UNIT = ***NO** / ***Y**ES
- ,**READ-ATTR**IBUTES = ***NO** / ***Y**ES
- ,**CHAN**GE-**ATTR**IBUTES = ***NO** / ***Y**ES
- ,**DEL**ETE-**FILE** = ***NO** / ***Y**ES

,**LEG**AL-**QUAL**IFICATION = ***UNCHA**NGED

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99.

FILE =
Name of the file in the remote FT partner system.

FILE = *NOT-SPECIFIED
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the file in the remote system. It must be specified in the syntax of the remote system and conform to the conventions of the remote system.
If the file name is specified with unattached Public Volume Set, the request is rejected with the error message FTR2202.

PASSWORD =
The password that provides access to the file in the remote system. If the file in the remote system is password-protected, the password required for modifying file attributes in remote systems must be specified in these operands.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
The password that provides access to the file in the remote system. The password must be in the syntax of the remote system and conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Contains the specifications for transfer admission to the remote system for file management requests.

TRANSFER-ADMISSION = *NONE
The remote system does not require or recognize any user authorization.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an admission profile if the FTAC functionality is in use in the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified. The alphanumeric entry is converted internally to lowercase characters.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the identification, the account number and the password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number for the user in the remote system. The account number must be specified in the syntax of the remote system and must observe its conventions.

PASSWORD =
The password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
The password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and be recognized by the remote system.

NEW-NAME =
New name of the file in the remote FT partner system.

NEW-NAME = *SAME
The previous file name remains unchanged.

NEW-NAME = <filename 1..54> / <c-string 1..512 with-low>
The new name of the file in the remote system. The previous name is no longer valid. The file name must be specified in the syntax of the remote system and conform to the conventions of the remote system.

FILE-AVAILABILITY = *UNCHANGED
The previous file availability remains unchanged.

STORAGE-ACCOUNT = *UNCHANGED
The previous account number remains unchanged.

FUTURE-FILE-SIZE = *UNCHANGED
The previous file size remains unchanged.

ACCESS-MODE =
Permitted access methods.

ACCESS-MODE = *UNCHANGED
The previous access rights remain unchanged.

ACCESS-MODE = *READ-ONLY
Short form of the current access rights READ-FILE, READ-ATTRIBUTES and CHANGE-ATTRIBUTE, and thus simplifies input.

ACCESS-MODE = *READ-WRITE
Short form of the current access rights READ-FILE, REPLACE-FILE, EXTEND-FILE, READ-ATTRIBUTES, CHANGE-ATTRIBUTES, DELETE-FILE, and ERASE-DATA, and thus simplifies input.

ACCESS-MODE = *REPLACE-ALL-BY(...)
The existing access rights of the file in the remote system are replaced by the specified access rights.

READ-FILE = *NO / *YES
The file cannot or can be read.

REPLACE-FILE = *NO / *YES
The file cannot or can be overwritten.

EXTEND-FILE = *NO / *YES
The file cannot or can be extended.

READ-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be read.

CHANGE-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be modified.

DELETE-FILE = *NO / *YES
The file cannot or can be deleted.

INSERT-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be inserted in the file.

ERASE-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be deleted from the file.

LEGAL-QUALIFICATION = *UNCHANGED
The previous legal qualifications remain unchanged.

Example

You wish to reset the access rights of the remote file MYFILE from READ-WRITE to READ-ONLY. The file is stored in the BS2000 system HUGO under the user ID JIM, with the account number A1234FT and the password C'PWD'

```
FTMOD PARTNER=HUGO,FILE-NAME=MYFILE,   -
                             TRANSFER-ADMISSION=(JIM,A1234FT,C'PWD'),-
                             ACCESS-MODE=*READ-ONLY
```

Short form:

```
FTMOD HUGO,MYFILE,,(JIM,A1234FT,'PWD'),,,,,*R-O
```

5.15 FTMODADS Modify admission set

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The FTAC user can modify the admission set for his/her own user ID with the FTMODADS command. You may access two components of the admission set:

a) You can define a password to be entered for almost all subsequent FTAC commands (except the FTSHW... commands). This prevents other users working with your user ID from entering FTAC commands.

It is not possible to have an FTAC password output. If an FTAC user forgets his/her FTAC password, only the FTAC administrator can delete or modify the password.

b) FTAC users may modify the limit values for the maximum number of security levels that can be reached from their user ID (the MAX-USER-LEVELS) within the range specified by the FTAC administrator. The limit values defined by the FTAC administrator (MAX-ADM-LEVELS) cannot, however, be overridden by the FTAC user. They can simply reduce the limit values since, in the case of FT requests, FTAC performs the admission check on the basis of the smallest value in the admission set. The MAX-USER-LEVELS are only effective if they are lower, i.e. more restrictive, than the MAX-ADM-LEVELS.

Format

FTMODADS

USER-IDENTIFICATION = ***OWN** / <name 1..8>

,**PASS**WORD = ***NONE** / <alphanum-name 1..8>

,**SEL**ECT-PARAMETER = ***ALL**

,**NEW-PASS**WORD = ***OLD** / ***NONE** / <alphanum-name 1..8>

,**PRIV**ILEGED = ***UNCHA**NGED

,**MAX-LEV**ELS = ***UNCH**ANGED / ***STD** / <integer 0...100> / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

OUTBOUND**-SEND** = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**OUT**BOUND**-REC**EIVE = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND**-SEND** = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND**-REC**EIVE = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND**-PROC**ESSING = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND**-MAN**AGEMENT = ***UNCH**ANGED / ***STD** / <integer 0...100>

Operands

USER-IDENTIFICATION =
User ID whose admission set is to be modified.

USER-IDENTIFICATION = *OWN
The admission set for the user ID which you are currently using is to be modified.

USER-IDENTIFICATION = <name 1..8>
The admission set for this user ID is to be modified. The FTAC user can only enter his/her own user ID here.

PASSWORD =
FTAC password which authorizes you to use FTAC commands, if such a password was defined in your admission set. An FTAC password is set with the operand NEW-PASSWORD.

PASSWORD = *NONE
No FTAC password is required for this admission set.

PASSWORD = <alphanum-name 1..8>
This password authorizes this user to use FTAC commands.

SELECT-PARAMETER = *ALL
In later openFT-AC versions it will be possible to specify additional selection criteria here.

NEW-PASSWORD =
Changes the FTAC password. If such an FTAC password has already been set, it must be used for almost all FTAC commands on the user ID for this admission set (except: the FTSHW... commands). This is done using the parameter PASSWORD in the respective commands.

NEW-PASSWORD = *OLD
The FTAC password remains unchanged.

NEW-PASSWORD = *NONE
No FTAC password is required for the user ID associated with this admission set.

NEW-PASSWORD = <alphanum-name 1..8>
Specification of the new FTAC password.

PRIVILEGED = *UNCHANGED
This parameter is only supported for reasons of compatibility.

MAX-LEVELS =
You set which security level(s) you can access, with which basic functions, from the user ID of this admission set. Either you can set one security level for all basic functions or different security levels for each basic function.
The MAX-USER-LEVELS for this admission set are set by the FTAC user; the MAX-ADM-LEVELS are set by the FTAC administrator.
FTAC runs authorization checks on the basis of the lowest specified security level. FTAC users may reduce but not increase the values specified for them by the FTAC administrator, see example to FTSHWADS.

MAX-LEVELS = *UNCHANGED
The security levels set in this admission set are to remain unchanged.

MAX-LEVELS = *STD
For this admission set, the values of the default admission set are valid. The admission set is deleted from the admission file. This is possible if the user ID has already been deleted.

MAX-LEVELS = <integer 0...100>
You can set a maximum security level for all six basic functions. The value 0 means that no file transfer is possible on this user ID until further notice (until the admission set is modified again).

MAX-LEVELS = *PARAMETERS(...)
You can set a maximum security level for each of the basic functions.

OUTBOUND-SEND =
Sets the maximum security level for the basic function “outbound send”. The owner of the admission set can send files to all partner systems whose security level has this value or lower.

OUTBOUND-SEND = *UNCHANGED
The value for OUTBOUND-SEND remains unchanged.

OUTBOUND-SEND = *STD
For OUTBOUND-SEND, the value from the default admission set is used.

OUTBOUND-SEND = <integer 0..100>
For OUTBOUND-SEND, this maximum security level is entered in the admission set.

OUTBOUND-RECEIVE =
Sets the maximum security level for the basic function “outbound receive”. The owner of the admission set can receive files from all partner systems whose security level has this value or lower.

OUTBOUND-RECEIVE = *UNCHANGED
The value for OUTBOUND-RECEIVE remains unchanged.

OUTBOUND-RECEIVE = *STD
For OUTBOUND-RECEIVE, the value from the default admission set is used.

OUTBOUND-RECEIVE = <integer 0..100>
For OUTBOUND-RECEIVE, this maximum security level is entered in the admission set.

INBOUND-SEND =
Sets the maximum security level for the basic function “inbound send”. All partner systems with this security level or lower can request files from the owner of the admission set.

INBOUND-SEND = *UNCHANGED
The value for INBOUND-SEND remains unchanged.

INBOUND-SEND = *STD
For INBOUND-SEND, the value from the default admission set is used.

INBOUND-SEND = <integer 0..100>
For INBOUND-SEND, this maximum security level is entered in the admission set.

INBOUND-RECEIVE =
Sets the maximum security level for the basic function “inbound receive”. All partner systems with this security level or lower may send files to the owner of the admission set.

INBOUND-RECEIVE = *UNCHANGED
The value for INBOUND-RECEIVE remains unchanged.

INBOUND-RECEIVE = *STD
For INBOUND-RECEIVE, the value from the default admission set is used.

INBOUND-RECEIVE = <integer 0..100>
For INBOUND-RECEIVE, this maximum security level is entered in the admission set.

INBOUND-PROCESSING =
Sets the maximum security level for the basic function “inbound processing”. All partner systems which have this security level or lower may include follow-up processing in their system as part of an FT request.

INBOUND-PROCESSING = *UNCHANGED
The value for INBOUND-PROCESSING remains unchanged.

INBOUND-PROCESSING = *STD
For INBOUND-PROCESSING, the value from the default admission set is used.

INBOUND-PROCESSING = <integer 0..100>
For INBOUND-PROCESSING, this maximum security level is entered in the admission set.

INBOUND-MANAGEMENT =
Sets the maximum security level for the basic function “inbound file management”. All partner systems with this security level or lower may include the modification of file attributes and the querying of directories as part of their FT request.

INBOUND-MANAGEMENT = *UNCHANGED
The value for INBOUND-MANAGEMENT remains unchanged.

INBOUND-MANAGEMENT = *STD
For INBOUND-MANAGEMENT, the value from the default admission set is used.

INBOUND-MANAGEMENT = <integer 0..100>
For INBOUND-MANAGEMENT, this maximum security level is entered in the admission set.

Example

Steven needs information on his admission sets.

```
FTSHWADS
```

He receives the following output:

```
               MAX. USER LEVELS                MAX. ADM LEVELS        ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF    OBS  OBR  IBS  IBR  IBP  IBF
 DACKTAIL 100  100  100  100  100  100     80   80   80   80   60   60
```

Steven forbids any follow-up processing and thus only allows FT functions.

```
FTMODADS MAX-LEVELS=*PARAMETERS(INBOUND-PROCESSING = 0)
```

The short form of this command is

```
FTMODADS MAX-LEV=(IN-PROC=0)
```

He outputs his admission set once more to double-check.

```
FTSHWADS
```

He receives the following output:

```
                MAX. USER LEVELS                MAX. ADM LEVELS          ATTR
USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
DACKTAIL 100  100  100  100    0  100   80   80   80   80   60   60
```

Although the FTAC administrator permitted follow-up processing (IBP) for all partners with a security level of 60 or lower, this is no longer possible on Steven's user ID. However, Steven then sets up a profile for trustworthy partners which allows them follow-up processing again.

```
FTCREPRF FRIENDS,TRANS-AD='for my friends',IGN-MAX-LEV=(IN-PROC=*YES), -
      PROCESSING-ADMISSION=(STEVEN,XXXX,PASSWORD)
```

5.16 FTMODDIR
Modify remote directory attributes

Note on usage

User group: FT user

Functional description

With the FTMODDIR command, you can modify the attributes of a directory in an FT partner system. It is currently only possible to change the directory name.

Format

FTMODDIR

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
,**ACCOUNT** = ***NONE** / <c-string 1..64 with-low> / <text 1..64>
,**PASS**WORD = ***NONE** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>

,**NEW-NAME** = ***SAME** / <filename 1..54> / <c-string 1..512 with-low>

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Defining the partner computer” on page 99.

DIRECTORY-NAME =
Name of the directory in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password permitting unrestricted access to the directory in the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password making it possible to access the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. The alphanumeric input is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

NEW-NAME =
New name of the directory in the remote FT partner system.

NEW-NAME = *SAME
The directory name is unchanged.

NEW-NAME = <filename 1..54> / <c-string 1..512 with-low>
New name of the directory in the remote system. The previous directory name becomes invalid. The directory name must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

Example

You wish to rename the PO library OTTO.CLIST to HUGO.CLIST on a remote z/OS system with the partner name ZOS2PART. The library is located under the account OPFT000:

```
FTMODDIR ZOS2PART,DIR-NAME='OPFT000.OTTO.CLIST'
        ,TRANS-ADM=(OPFT000,ACCT,PASSWORD)
        ,NEW-NAME=hugo.clist
```

5.17 FTMODPRF Modify admission profile

User instruction

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The command FTMODPRF can be used by any FTAC user to modify his/her admission profile. In a privileged admission profile, an FTAC user can only modify the operands TRANSFER-ADMISSION and PRIVILEGED.

As soon as an admission profile is modified, the timestamp of the last modification is also updated. You can see the timestamp with FTSHWPRF INF=*ALL (LAST-MODIF). The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF with the parameter NAME without specifying other parameters.

Format

(part 1 of 2)

FTMODPRF

NAME = ***ALL** / ***STD** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <alphanum-name 1..8>

,SELECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
TRANSFER-ADMISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> / c-string 8..32 with-low> / <x-string 15..64>
,OWNER-IDENTIFICATION = ***OWN** / <name 1..8>

,NEW-NAME = ***OLD** / ***STD** / <alphanum-name 1..8>

,TRANSFER-ADMISSION = ***UNCH**ANGED / ***NOT-SPEC**IFIED / ***OLD-ADM**ISSION(...) / <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)

***OLD-ADM**ISSION(...)
VALID = ***UNCH**ANGED / ***YES** / ***NO**
,USAGE = ***UNCH**ANGED / ***PRIVATE** / ***PUB**LIC
,EXPIRATION-DATE = ***UNCH**ANGED / ***NOT-REST**RICTED / <date 8..10>

<alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
VALID = ***YES** / ***NO** / ***UNCH**ANGED
,USAGE = ***PRIVATE** / ***PUB**LIC / ***UNC**HANGED
,EXPIRATION-DATE = ***NOT-REST**RICTED / <date 8..10> / ***UNCH**ANGED

,PRIVILEGED = ***UNCH**ANGED / ***NO**

,IGNORE-MAX-LEVELS = ***UNCH**ANGED / ***NO** / ***YES** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
OUTBOUND-SEND = ***UNCH**ANGED / ***NO** / ***YES**
,**OUTBOUND-RECEIVE** = ***UNCH**ANGED / ***NO** / ***YES**
,**INBOUND-SEND** = ***UNCH**ANGED / ***NO** / ***YES**
,**INBOUND-RECEIVE** = ***UNCH**ANGED / ***NO** / ***YES**
,**INBOUND-PROCESSING** = ***UNCH**ANGED / ***NO** / ***YES**
,**INBOUND-MANAGEMENT** = ***UNCH**ANGED / ***NO** / ***YES**

,USER-ADMISSION = ***UNCH**ANGED / ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = ***OWN** / <name 1..8>
,**ACCOUNT** = ***OWN** / ***NOT-SPEC**IFIED / ***NONE** / <alphanum-name 1..40> / <c-string 1..40>
,**PASS**WORD = ***OWN** / <alphanum-name 1..8> / ***NONE**

(part 2 of 2)

,INITIATOR = ***UNCH**ANGED / list-poss(2): ***REM**OTE / ***LOC**AL

,TRANSFER-DIRECTION = ***UNCH**ANGED / ***NOT-REST**RICTED / ***FROM**-PARTNER / ***TO**-PARTNER

,PARTNER = ***UNCH**ANGED / ***NOT-REST**RICTED / ***ADD**(...) / ***REM**OVE(...) /
list-poss(50): <text 1..200 with-low>

***ADD**(...)
NAME = list-poss(50): <text 1..200 with-low>

***REM**OVE(...)
NAME = list-poss(50): <text 1..200 with-low>

,MAX-PARTNER-LEVEL = ***UNCH**ANGED / ***NOT-REST**RICTED / <integer 0..100>

,FILE-NAME = ***UNCH**ANGED / ***NOT-REST**RICTED / <filename1..59> / <c-string 1..512 with-low> /
***EXP**ANSION(...)

***EXP**ANSION(...)
PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>

,FILE-PASSWORD = ***UNCH**ANGED / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

,PROCESSING-ADMISSION = ***UNCH**ANGED / ***SAME** / ***NOT-REST**RICTED / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = ***SAME** / ***NOT-REST**RICTED / <name 1..8>
,**ACC**OUNT = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..40> / <c-string 1..40>
,**PASSW**ORD = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

,SUCCESS-PROCESSING = ***UNCH**ANGED / ***NOT-RES**TRICTED / ***NONE** / <c-string 1..1000 with-low> /
***EXPA**NSION(...)

***EXP**ANSION(...)
PREFIX = ***UNCH**ANGED / ***NOT-REST**RICTED / <c-string 1..999 with-low>
,SUFFIX = ***UNCH**ANGED / ***NOT-REST**RICTED / <c-string 1..999 with-low>

,FAILURE-PROCESSING = ***UNCH**ANGED / ***NOT-RES**TRICTED / ***NONE** / <c-string 1..1000 with-low> /
***EXPA**NSION(...)

***EXP**ANSION(...)
PREFIX = ***UNCH**ANGED / ***NOT-REST**RICTED / <c-string 1..999 with-low>
,SUFFIX = ***UNCH**ANGED / ***NOT-REST**RICTED / <c-string 1..999 with-low>

,WRITE-MODE = ***UNCH**ANGED / ***NOT-REST**RICTED / ***NEW**-FILE / ***REP**LACE-FILE / ***EXT**END-FILE

,FT-FUNCTION = ***UNCH**ANGED / ***NOT-REST**RICTED / list-poss(5):
***TRANS**FER-**F**ILE / ***MOD**IFY-**F**ILE-**ATTR**IBUTES / ***READ-DIR**ECTORY /
***FILE-PROC**ESSING

,USER-INFORMATION = ***UNCH**ANGED / ***NONE** / <c-string 1..100 with-low>

,DATA-ENCRYPTION = ***UNCH**ANGED / ***NOT-REST**RICTED / ***NO** / ***YES**

Operands

NAME =
Determines the name of the admission profile to be modified.

NAME = *ALL
Modifies all your admission profiles at the same time provided no further selection criteria are specified using the SELECT parameter and neither the name nor the transfer admission is to be modified.

NAME = *STD
Changes the default admission profile for your user ID.

NAME = <alphanum-name 1..8>
Modifies the admission profile with this name.

PASSWORD =
FTAC password which authorizes you to use FTAC commands on your user ID, if such a password has been defined in your admission set.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

SELECT-PARAMETER =
Specifies a transfer admission. You will then modify the admission profile which has this transfer admission.

SELECT-PARAMETER = *OWN
Modifies your own admission profile.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies the selection criteria for the profiles which you wish to modify.

TRANSFER-ADMISSION =
Entering the TRANSFER-ADMISSION here makes it a selection criterion for the admission profiles which you wish to modify.

TRANSFER-ADMISSION = *ALL
All your admission profiles are to be modified, irrespective of the transfer admission.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Only admission profiles without a defined transfer admission are to be modified. In the case of a default admission profile, the transfer admission is never assigned, because this is addressed using the user ID and the user password.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The admission profile with this transfer admission is to be modified.

OWNER-IDENTIFICATION =
You can use the owner of an admission profile as a selection criterion for access to a profile to be modified.

OWNER-IDENTIFICATION = *OWN
Modifies your own admission profile.

OWNER-IDENTIFICATION = <name 1..8>
The FTAC user can enter only his/her own user ID here, the FTAC administrator can enter any user ID.

NEW-NAME =
NEW-NAME is used to assign a new name to the admission profile.
NEW-NAME may only be specified together with unambiguous selection criteria (NAME or TRANSFER-ADMISSION).

NEW-NAME = *OLD
The name of the admission profile remains unchanged.

NEW-NAME = *STD
Makes the admission profile the default admission profile for the user ID. If the admission profile previously had a transfer admission, you must also specify TRANSFER-ADMISSION=*NOT-SPECIFIED.

NEW-NAME = <alphanum-name 1..8>
New name of the admission profile. This name must be unique among all the admission profiles on your user ID. If an admission profile with this name already exists, FTAC rejects the command with the following message:

```
FTC0100   COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 294ff) can be used to obtain information on the already existing name. For this information, it suffices to enter FTSHWPRF without parameters.

TRANSFER-ADMISSION =
Modifies the transfer admission which is associated with the admission profile selected. You must ensure that the transfer admission is unique within your openFT system. If the transfer admission which you have selected already exists, FTAC rejects the command with the following message:

```
FTC0101   COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

TRANSFER-ADMISSION may only be specified together with unambiguous selection criteria (NAME or SELECT-PARAMETERS=*PAR(TRANSFER-ADMISSION).

TRANSFER-ADMISSION = *UNCHANGED
The transfer admission remains unchanged.

TRANSFER-ADMISSION = *NOT-SPECIFIED
No transfer admission is set and any existing transfer admissions are made invalid. This blocks the profile, provided that it is not a profile that you are converting to a default admission profile. In this case, you must specify *NOT-SPECIFIED.

TRANSFER-ADMISSION = *OLD-ADMISSION(...)
The transfer admission itself remains unchanged. The options, however, can be changed, as opposed to with the entry TRANSFER-ADMISSION=*UNCHANGED. The specifications are ignored if you are changing a default admission profile.

VALID = *UNCHANGED
The value remains unchanged.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be blocked with this entry.

USAGE = *UNCHANGED
The value remains unchanged.

USAGE = *PRIVATE
Access to your profile is denied for security reasons whenever another user ID attempts to set for a second time the TRANSFER-ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for uniqueness reasons.

EXPIRATION-DATE = *UNCHANGED
The value remains unchanged.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
Date in the form *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2013-03-31 or 13-03-31 for 31 March, 2013.The use of the transfer admission is only possible until the given date.

TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
The character string must be entered as transfer admission in the transfer request. The alphanumeric input is always stored in lowercase letters.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be blocked with this entry.

VALID = *UNCHANGED
The value remains unchanged.

USAGE = *PRIVATE
Access to your profile is denied for security reasons whenever another user ID attempts to set for a second time the TRANSFER-ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for uniqueness reasons.

USAGE = *UNCHANGED
The value remains unchanged.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
Date in the form *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2013-03-31 or 13-03-31 for 31 March, 2013.The use of the transfer admission is only possible until the given date.

EXPIRATION-DATE = *UNCHANGED
The value remains unchanged.

PRIVILEGED =
The FTAC administrator can privilege the admission profile of any FTAC user. FT requests which are processed with a privileged status are not subject to the restrictions for MAX-ADM-LEVEL in the admission set.
The FTAC user can only reverse any privileged status given.

PRIVILEGED = *UNCHANGED
The status of this admission profile remains unchanged.

PRIVILEGED = *NO
With *NO, you can reverse the privileged status.

IGNORE-MAX-LEVELS =
Determines for which of the six basic functions the restrictions of the admission set should be ignored. The user’s MAX-USER-LEVELS can be exceeded in this way. The MAX-ADM-LEVELS in the admission set can only be effectively exceeded with an admission profile which has been designated as privileged by the FTAC administrator. The FTAC user can set up an admission profile for himself/herself for special tasks (e.g. sending a certain file to a

partner system with which he/she normally is not allowed to conduct a file transfer), which allows him/her to exceed the admission set. This profile must be explicitly given privileged status by the FTAC administrator.

If you enter IGNORE-MAX-LEVELS=*YES, the settings for all the basic functions are ignored. If you wish to ignore the admission set for specific basic functions, you need to do this with the operands explained later in the text.

The following table shows which partial components of the file management can be used under which conditions:

Inbound file management function	**Setting in admission set/extension in profile**
Show file attributes	Inbound sending (IBS) permitted
Modify file attributes	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Rename files	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Delete files	Inbound receiving (IBR) permitted **and** write rule = overwrite in profile
Show directories	Inbound file management (IBF) permitted **and** direction = to partner in profile
Create, rename, delete directories	Inbound file management (IBF) permitted **and** direction = from partner in profile

IGNORE-MAX-LEVELS = *UNCHANGED
You can access the same security levels as before the modification (unless you have reversed the privileged status with PRIVILEGED=*NO).

IGNORE-MAX-LEVELS = *NO
FT requests which are processed with the admission profile are subject to the restrictions of the admission set.

IGNORE-MAX-LEVELS = *YES
*YES allows you to communicate with partner systems whose security level exceeds the specifications of the admission set. If your profile does not have privileged status, you can only disregard the MAX-USER-LEVELS in the admission set, not the MAX-ADM-LEVELS. The current MAX-USER-LEVELS and MAX-ADM-LEVELS settings can be accessed using the command SHOW-FT-ADMISSION-SET (see example on page 256).

IGNORE-MAX-LEVELS = *PARAMETERS(...)

OUTBOUND-SEND = *UNCHANGED
The maximum security level which can be reached with the basic function “outbound send” remains unchanged.

OUTBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “outbound send” is determined by the admission set.

OUTBOUND-SEND = *YES
For the basic function “outbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

OUTBOUND-RECEIVE = *UNCHANGED
The maximum security level which can be reached with the basic function “outbound receive” remains unchanged.

OUTBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “outbound receive” is determined by the admission set.

OUTBOUND-RECEIVE = *YES
For the basic function “outbound receive”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-SEND = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound send” remains unchanged.

INBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “inbound send” is determined by the admission set.

INBOUND-SEND = *YES
For the basic function “inbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial component “display file attributes” of the basic function “inbound file management” can be used.

INBOUND-RECEIVE = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound receive” remains unchanged.

INBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “inbound receive” is determined by the admission set.

INBOUND-RECEIVE = *YES
Disregards your settings for “inbound receive” in the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

The same applies to the following partial components of the basic function “inbound file management”:

– delete files, as long as the file attributes are set accordingly,
– modify file attributes, if the basic function “inbound file management” was admitted in the admission set or in the admission profile.

INBOUND-PROCESSING = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound processing” remains unchanged.

INBOUND-PROCESSING = *NO
The maximum security level which can be reached with the basic function “inbound processing” is determined by the admission set.

INBOUND-PROCESSING = *YES
For the basic function “inbound processing”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-MANAGEMENT = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound file management” remains unchanged.

INBOUND-MANAGEMENT = *NO
The maximum security level which can be reached with the basic function “inbound file management” is determined by the admission set.

INBOUND-MANAGEMENT = *YES
For the basic function “inbound file management”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The partial component “modify file attributes” of the basic function “inbound file management” only functions if the basic function “inbound receive” was admitted in the admission set or admission profile.

USER-ADMISSION =
User ID under which the modified admission profile is saved. FT requests which use this profile access the entered user ID in the local system.
As an FTAC user you can only specify your own user ID here.
If the FTAC administrator has created an admission profile for a user without specifying the access data (see the FTCREPRF command in the openFT System Asministrator Guide), the user must, if necessary, enter the account and password in the operands ACCOUNT and PASSWORD described below before the profile can be used.

USER-ADMISSION = *UNCHANGED
The USER-ADMISSION of this admission profile remains unchanged.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications are taken from the current LOGON authorization. A z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile.

Admission profiles in which USERID, ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, these parameters must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

USER-ADMISSION = *PARAMETERS(...)
Specifies the individual components of the user ID.
This allows you, for example, to ensure that FT requests which use this admission profile are kept under a different account number from the currently valid account number. Another application is to specify a passwrod in the admission profile. FT requests which use this admission profile will then only function if the current LOGON password corresponds to this preset password.

USER-IDENTIFICATION =
Your user ID in z/OS

USER-IDENTIFICATION = *OWN
The user ID is taken from your LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID with which the profile is to be associated.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = *OWN
The account number is taken from the current LOGON authorization.

ACCOUNT = *NOT-SPECIFIED
No account number is defined.
The account number is to be specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for user IDs whose account numbers he/she does not know.
For further details, see “Default account number” on page 126.

ACCOUNT = *NONE
Has the same effect as ACCOUNT = *NOT-SPECIFIED.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which is associated with your user ID.

PASSWORD =
Password which an FT request is to use when it works with this admission profile.

PASSWORD = *OWN
When an FT request refers to this admission profile, FTAC uses the password valid at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.
Admission profiles in which PASSWORD is set to its default value via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, this parameter must be explicitly assigned a value. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

PASSWORD = *NOT-SPECIFIED
The password is specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for foreign user IDs.

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the specified password is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

PASSWORD = *NONE
No password is required for the user ID.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = *UNCHANGED
The settings in this admission profile remain unchanged,

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile.


The transfer direction is always determined from the system in which the admission profile was defined.

TRANSFER-DIRECTION = *UNCHANGED
The specification in the admission profile remains unchanged.

TRANSFER-DIRECTION = *NOT-RESTRICTED
Files can be transferred to and from a partner system.

TRANSFER-DIRECTION = *FROM-PARTNER
Files can only be transferred from a partner system to your system. It is not possible to display file attributes/directories (partial components of “inbound file management”).

TRANSFER-DIRECTION = *TO-PARTNER
Files can only be transferred from your system to a partner system. It is not possible to modify file attributes or delete files (partial components of “inbound file management”).

PARTNER =
Specifies that this admission profile is to be used only for FT requests which are processed by a a certain partner system.

PARTNER = *UNCHANGED
Any partner in the admission profile remains unchanged.

PARTNER = *NOT-RESTRICTED
This admission profile’s scope of use is not limited to FT requests with certain partner systems.

PARTNER = *ADD(NAME = list-poss(50): <text 1..200 with-low>)
With this specification, you can add elements to an existing list of partner systems.
A maximum of 50 partner systems can be specified.

PARTNER = *REMOVE(NAME = list-poss(50): <text 1..200 with-low>)
Removes elements from an existing list of partner systems. A maximum of 50 partner systems can be specified.

PARTNER = list-poss(50): <text 1..200 with-low>
The admission profile only permits those FT requests which are processed with the specified partner systems. A maximum of 50 partner systems can be specified.
For PARTNER you can specify the name from the partner list or the address of the partner system, see also section “Defining the partner computer” on page 99. You are advised to use the name from the partner list.

MAX-PARTNER-LEVEL =
A maximum security level can be specified. The admission profile will then only permit those FT requests which are processed with partner systems which have this security level or lower.
MAX-PARTNER-LEVEL works in conjunction with the admission set. When non-privileged admission profiles are used, the access check is executed on the basis of the smallest specified value.

MAX-PARTNER-LEVEL = *UNCHANGED
The specification for MAX-PARTNER-LEVEL in this admission set remains unchanged.

MAX-PARTNER-LEVEL = *NOT-RESTRICTED
If FT requests are processed with this admission profile, then the highest accessible security level is determined by the admission set.

MAX-PARTNER-LEVEL = <integer 0..100>
All partner systems which have this security level or lower can be communicated with.

When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the time being). No FT request can then be processed with this admission profile.

FILE-NAME =
Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *UNCHANGED
The specifications for FILE-NAME in this admission profile remain unchanged.

FILE-NAME = *NOT-RESTRICTED
The admission profile permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Only the specified file may be accessed. However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here (see section “Unique file names for receive files” on page 52). When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX, see page 116. You can also directly specify file transfer with pre- and post-processing here by entering the pipe symbol ’|’ followed by a command.

FILE-NAME =*EXPANSION(PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>)
Restricts access to a number of files which all begin with the same prefix. If a *filename* is entered in an FT request which uses this admission profile, FTAC sets the *prefix* defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example

– PREFIX=STEVEN.; An FT request in which the FILE-NAME=MILLER is specified accesses the file STEVEN.MILLER.

– PREFIX=TOOLS.CLIST/; an FT request in which FILE-NAME=MEMBER01 was specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or post-processing, you should indicate this by entering the pipe symbol ‘|’ at the start of the prefix. The created FTAC profile can then be used only for file transfer with pre- or post-processing since the file name that is generated also starts with a ‘|’. The variable %TEMPFILE can also be used in the filename prefix. You can find detailed information on preprocessing and postprocessing in section “Preprocessing and postprocessing” on page 111.

The maximum length of the entire pre- or post-processing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

– In the case of admission profiles which are to be used exclusively for the ftexec command you must specify a filename or filename prefix that starts with the character string ‘|ftexecsv’ (see FTCREPRF, “Example 3” on page 186).

– Specify the file name prefix '|*ftmonitor' for admission profiles that are exclusively used for monitoring. A profile of this sort can then be used in the openFT Monitor or in an ft or ncopy command from a Windows or Unix system (see “Example 2” on page 186).

FILE-PASSWORD =
You can enter a password for files into the admission profile. The FTAC functionality then only permits access to files which are protected with this password and to unprotected files. When a FILE-PASSWORD is specified in an admission profile, the password may no longer be specified in an FT request which uses this admission profile. This allows you to permit access to certain files to users in remote systems, without having to disclose the file passwords.

FILE-PASSWORD = *UNCHANGED
The specifications for FILE-PASSWORD in this admission profile remain unchanged.

FILE-PASSWORD = *NOT-RESTRICTED
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *UNCHANGED
The PROCESSING-ADMISSION in this admission profile remains unchanged.

PROCESSING-ADMISSION = *SAME
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION= *NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows follow-up processing using this admission profile and started from FT requests to be charged under a different account number, for example. Or, a password can be set in the admission profile. Follow-up processing for FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
User ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = *SAME
The USER-IDENTIFICATION is taken from the USER-ADMISSION.

USER-IDENTIFICATION = *NOT-RESTRICTED
The admission profile does not restrict the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = <name 1..8>
FT requests which are processed with this admission profile are only permitted follow-up processing under this user ID. If another user ID is entered here, the parameter PASSWORD must also be entered. PASSWORD=*SAME is then not valid.

ACCOUNT =
Specifies the account number for the follow-up processing.

ACCOUNT = *SAME
The account number is taken from the USER-ADMISSION.

ACCOUNT = *NOT-RESTRICTED
The account number may be specified in FT requests that work with the admission profile. The admission profile does not restrict the account for follow-up processing.

ACCOUNT = *NONE
The account number is used which is defined as the default account number of the user ID specified at the time the admission profile is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You can also specify account information containing the account number to be used.

PASSWORD =
Specifies, where applicable, the z/OS password for the user ID under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn't have such a password (yet).

PASSWORD = *SAME
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the BS2000 password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
The password may be specified for FT requests which work with the admission profile. The admission profile does not restrict the password for follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use the admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = *UNCHANGED
The specifications for SUCCESS-PROCESSING in this admission profile remain unchanged.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
BS2000 commands which are executed in the local system after successful data transfer. The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (' or ") within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *UNCHANGED
The specifications for the follow-up processing prefix in this admission profile remain unchanged.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *UNCHANGED
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC='"FILE TRANSFER OK"' is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = *UNCHANGED
The specifications for FAILURE-PROCESSING in this admission profile remain unchanged.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
z/OS commands which are executed in the local system after failed data transfer. Individual commands must be preceded by a slash (/). The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *UNCHANGED
The specifications for the follow-up processing prefix in this admission profile remain unchanged.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *UNCHANGED
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE which is valid for this FT request. WRITE MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = *UNCHANGED
The specifications for WRITE-MODE in this admission profile remain unchanged.

WRITE-MODE = *NOT-RESTRICTED
In an FT request which accesses this admission profile, WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *EXTEND-FILE may be entered for WRITE-MODE.

FT-FUNCTION =
This operand permits the restriction of the profile validity to certain FT functions (=file transfer and file management functions), see also page 40.

FT-FUNCTION = *UNCHANGED
The previous scope of the FT functions remains unchanged.

FT-FUNCTION = *NOT-RESTRICTED
The full scope of FT functions is available ..

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *FILE-PROCESSING)
The following file transfer functions are available:

***TRANSFER-FILE**
The admission profile may be used for the file transfer functions “transfer files”, “view file attributes” and “delete files”.

***MODIFY-FILE-ATTRIBUTES**
The admission profile may be used for the file transfer functions “view file attributes” and “modify file attributes”.

***READ-DIRECTORY**
The admission profile may be used for the file transfer functions “view directories” and “view file attributes”.

***FILE-PROCESSING**
The admission profile may be used for the “pre-processing” and “post-processing” file transfer functions. The “transfer files” function must also be permitted.

The *FILE-PROCESSING specification is of relevance only for FTAC profiles without a filename prefix. Otherwise the first character of the filename prefix determines whether only normal data transfer (no pipe symbol “|”) or only pre- and post-processing (pipe symbol “|”) are to be possible with this FTAC profile.

USER-INFORMATION =
Specifies a text in the admission profile. This text can be displayed with the FTSHWPRF command.

USER-INFORMATION = *UNCHANGED
Any existing text remains unchanged.

USER-INFORMATION = *NONE
Any existing text is deleted.

USER-INFORMATION = <c-string 1..100 with-low>
The character string entered is accepted as user information.

DATA-ENCRYPTION =
Specifies whether user data with this profile must be transferred in encrypted form.

DATA-ENCRYPTION = *UNCHANGED
The encryption option should remain unchanged.

DATA-ENCRYPTION = *NOT-RESTRICTED
The encryption option for user data is not restricted. File transfer requests with encryption and file transfer requests without encryption are both accepted

DATA-ENCRYPTION = *NO
Only file transfer requests that do not have encrypted user data are accepted, i.e. requests with encryption are rejected. If the request is made in a BS2000 or z/OS, DATA-ENCRYPTION=*NO must be specified there in the NCOPY request.

DATA-ENCRYPTION = *YES
Only file transfer requests that have encrypted user data are accepted, i.e. requests without encryption are rejected. If the request is made in a BS2000 or z/OS, for example, then DATA-ENCRYPTION=*YES must be specified there in the NCOPY request.

When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP('PR DSNAME(XYZ.',')')

Example

After Steven Miller has created an admission profile with the name *profile1*, which permits other users access to his user ID with the LOGON authorization, he decides he wants to restrict this profile so that only FT accesses are possible to files which begin with the prefix *BRANCH*.

The required command is:

```
FTMODPRF␣NAME=PROFIL1,
         FILE-NAME=*EXPANSION(PREFIX=BRANCH.)
```

A possible short form of this command is:

```
FTMODPRF␣PROFIL1,FILE-N=(PRE=BRANCH.)
```

This places heavy restrictions on the admission profile. The other specifications remain unchanged.

5.18 FTMODREQ Modify request queue

Note on usage

User group: FT user and FT administrator

Functional description

You use the FTMODREQ command to modify the position and priority of your outbound requests within the openFT request queue. You have the option of processing the outbound requests in any order you wish. Newly input requests or requests whose priority changes are put at the end of the request queue for the corresponding priority. If already active requests are repositioned behind waiting outbound requests, the active requests are interrupted if possible in favor of those waiting.

FTMODREQ is only valid for outbound requests.

The sequence of requests with a starting time in the future cannot be modified.

As FT user you can only modify your own requests.

Format

FTMODREQ

TRANSFER-ID = **<u>*ALL</u>** / <integer 1..2147483647>

,**SEL**ECT = **<u>*OWN</u>** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
| **OWN**ER-**ID**ENTIFICATION = **<u>*OWN</u>** / <name 1..8>
| ,**PART**NER = **<u>*ALL</u>** / <text 1..200 with-low>
| ,**FILE** = **<u>*ALL</u>** / <filename 1..59> / <c-string 1..512 with-low>

,**QUEUE-POS**ITION = **<u>*UNCHANGED</u>** / ***FIRST** / ***LAST**

,**PRIO**RITY = **<u>*UNCHANGED</u>** / ***NORMAL** / ***HIGH** / ***LOW**

Operands

TRANSFER-ID =
Transfer ID of the outbound request to be modified.

TRANSFER-ID = <u>*ALL</u>
Modifies all outbound requests, If further selections haven't been specified with SELECT (see below). FT users can only modify requests under their own user ID.

TRANSFER-ID = <integer 1..2147483647>
Transfer ID which is communicated to the local system in the FT request confirmation.

SELECT =
Contains selection criteria for outbound requests to be modified. A request is only modified if all the criteria specified are met.

SELECT = *OWN
Modifies all FT requests of the user's own ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
Identifies the owner of the FT request.

OWNER-IDENTIFICATION = *OWN
Modifies only outbound requests with the user's own ID.

OWNER-IDENTIFICATION = <name 1..8>
Specifies a user ID whose requests are to be modified.
Users may only enter their own user ID.

PARTNER =
Modifies outbound requests which are to be executed with a particular partner system.

PARTNER = *ALL
The name of the partner system is not selected as a criterion for the outbound requests to be modified.

PARTNER = <text 1..200 with-low>
Modifies outbound requests which are to be executed with this partner system. You can specify the name from the partner list or the address of the partner system. For more information on address specifications, see section "Defining the partner computer" on page 99.

FILE =
Modifies outbound requests which access this file or library member in the local system as a send or receive file. The file or library member name must be entered exactly as in the file transfer request and as it is output using the NSTATUS command. File names with wildcards are not permitted.

FILE = *ALL
The filename is not selected as a criterion for the outbound requests to be modified.

FILE = <filename 1..59> / <c-string 1..512 with-low>
Modifies outbound requests which access this file (DVS/POSIX) in the local system.

QUEUE-POSITION =
New position of the outbound request that is to be modified in the openFT request queue.

QUEUE-POSITION = *UNCHANGED
The position of the outbound request in this user's openFT request queue remains unchanged.

QUEUE-POSITION = *FIRST
The outbound request is placed in front of all the other requests of the same priority issued by the user in the openFT request queue.

QUEUE-POSITION = *LAST
The outbound request is placed behind all the other requests of the same priority issued by the user in the openFT request queue.

PRIORITY =
Modifies the priority of the FT request.

PRIORITY = *UNCHANGED
The priority of the FT request remains unchanged.

PRIORITY = *NORMAL
The priority of the FT request is set to the normal value

PRIORITY = *HIGH
The FT request is given a high priority.

PRIORITY = *LOW
The FT request is given a low priority.

Example

```
NSTATUS
  TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
  54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
  11164324  LOC   WAIT  UNIX2    FROM  0          FILE2

FTMODREQ SELECT=(FILE=FILE2),QUEUE-POS=*FIRST

NSTATUS
  TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
  11164324  LOC   WAIT  UNIX2    FROM  0          FILE2
  54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
```

5.19 FTSCOPY Transfer file synchronously

Note on usage

User group: FT user

Functional description

With the FTSCOPY command, you issue a synchronous request to send one or more files to the remote system or to retrieve one or more files from the remote system.

With a few exceptions, the operands are identical to those of the NCOPY command. Consequently only the syntax is described.

FTSCOPY differs from NCOPY in the following points:

– There is no local follow-up processing. The local parameters PROCESSING-ADMISSION, SUCCESS-PROCESSING and FAILURE-PROCESSING are therefore omitted.

– The general parameters PRIORITY, START and CANCEL are not used, because they do not have any significance for synchronous transfer.

Format

(part 1 of 4)

FTSCOPY

TRANSFER-**DIR**ECTION = **TO-**PARTNER / **FROM-**PARTNER

,**PART**NER = <text 1..200 with-low>

,**LOC**AL-PARAMETER = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

FILE-NAME =***NOT-SPECIFIED** /<filename 1..59> / <c-string 1..512 with-low>

,**PASS**WORD = ***NONE** / <alphanum-name 1..8>

,**TRANS**FER-**ADM**ISSION = ***SAME** / <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low> / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = <name 1..8>

,**ACCOUNT** = ***NONE** / <alphanum-name 1..40> / c-string 1..40>

,**PASS**WORD = ***NONE** / <alphanum-name 1..8>

,**CODED-CHAR**ACTER-**SET** = ***STD** / <alphanum-name 1..8>

(part 2 of 4)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)

  *BS2000(...)
    FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <c-string 1..512 with-low>/
                  *LIBRARY-ELEMENT(...)
      *LIBRARY-ELEMENT(...)
        LIBRARY = *NOT-SPECIFIED / <filename 1..54>
        ,ELEMENT = *NOT-SPECIFIED /  <filename 1..64 without-gen-vers>(...) /
                     <composed-name 1..64 with-under>(...) / <number 1..ffff>
                   <filename>(...) / <composed-name>(...)
                     VERSION = *STD / <text 1..24>
        ,TYPE = *NOT-SPECIFIED / <name 1..8>
    ,PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> /
                  <integer -2147483648..2147483647>
    ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /
                            <c-string 8..32 with-low>/ *PARAMETERS(...)
      *PARAMETERS(...)
        USER-IDENTIFICATION = <name 1..8>
        ,ACCOUNT = *NONE / <alphanum-name 1..8>
        ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
    ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
      *PARAMETERS(...)
        USER-IDENTIFICATION = <alphanum-name 1..8>
        ,ACCOUNT = *NONE / <alphanum-name 1..8>
        ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
    ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
    ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
    ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 3 of 4)

```
*MSP(...)
    FILE-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low>
   ,PASSWORD = *NONE / <alphanum-name 1..8>
   ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low> /
                         *PARAMETERS(...)
      *PARAMETERS(...)
         USER-IDENTIFICATION = <name 1..8>
        ,ACCOUNT = *NONE<alphanum-name 1..40> / <c-string 1..40>
        ,PASSWORD = *NONE / <alphanum-name 1..8>
   ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
      *PARAMETERS(...)
         USER-IDENTIFICATION = <name 1..8>
        ,ACCOUNT = *NONE<alphanum-name 1..40> / <c-string 1..40>
        ,PASSWORD = *NONE / <alphanum-name 1..8>
   ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
   ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
   ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
*ANY(...)
    FILE-NAME = *NOT-SPECIFIED / <c-string 1..512 with-low> / *LIBRARY-ELEMENT(...)
      *LIBRARY-ELEMENT(...)
         LIBRARY = *NOT-SPECIFIED / <c-string 1..63 with-low>
        ,ELEMENT = *NOT-SPECIFIED / <c-string 1..64 with-low>(...)
            <c-string 1..64 with-low>(...)
               VERSION = *NONE / *STD / <c-string 1..24 with-low>
        ,TYPE = *NONE / *NOT-SPECIFIED / <c-string 1..8 with-low>
   ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
   ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                         <c-string 8..32 with-low> / *PARAMETERS(...)
      *PARAMETERS(...)
         USER-IDENTIFICATION = <c-string 1..67 with-low>
        ,ACCOUNT = *NONE / <c-string 1..64 with-low>
        ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
   ,PROCESSING-ADMISSION = *SAME / *NONE / *PARAMETERS(...)
      *PARAMETERS(...)
         USER-IDENTIFICATION = <c-string 1..67 with-low>
        ,ACCOUNT = *NONE / <c-string 1..64 with-low>
        ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
   ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
   ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
   ,CODED-CHARACTER-SET = *STD / <c-string 1..8 with-low>
```

(part 4 of 4)

,**COMP**RESS = ***NONE** / ***BYTE-REPET**ITION / ***ZIP**

,**WRITE**-MODE = ***REPLACE**-FILE / ***NEW-**FILE / ***EXT**END**-**FILE

,**DATA**-TYPE = ***NOT-SPECIFIED** / ***CHAR**ACTER (...) / ***BIN**ARY (...) / ***USER**

***CHAR**ACTER(...)
TRANSPARENT = ***NO** / ***YES**

***BIN**ARY(...)
TRANSPARENT = ***NO** / ***YES**

,**DATA-ENC**RYPTION = ***NO** / ***YES** / ***ONLY-DATA-INTEGRITY**

,**REC**ORD**-SIZE** = ***NOT-SPECIFIED** /<integer 1..32756>

,**REC**ORD**-FORM**AT = ***STD** / ***FIXED** / ***VARIABLE** / ***UNDEFINED**

,TABULATOR = ***AUTO** / ***ON** / ***OFF**

,TARGET-FILE-FORMAT = ***SAME** / ***BLOCK**-ORIENTED / ***SEQ**UENTIAL(...)

***SEQ**UENTIAL(…)
RECORD**-FORM**AT = ***SAME** / ***UNDEFINED**

Operands

The meaning of the operands is the same as for asynchronous file transfer, see the operand description for NCOPY as of page 319.

Example

The file EXAMPLE is to be transferred to the remote Unix system PUX. Here, it is to be stored in the directory dir (subdirectory of the HOME directory) under the transfer admission ForUXSys. ZIP compression is to be used for transfer.

```
FTSCOPY TRANS-DIR=TO,PARTNER=PUX,   -
*LOCAL=*PAR(FILE-NAME=EXAMPLE),          -
*REM=*ANY(FILE-NAME='dir/file.ux',TRANS='ForUXSys'), -
COMP=*ZIP

FTR0005 OPENFT: Request 91339. File 'EXAMPLE' transferred
```

Short form:

```
FTSCOPY TO,PUX,(EXAMPLE),*a('dir/file.ux',,'ForUXSys'),*ZIP
```

5.20 FTSHW Display remote file attributes

Note on usage

User group: FT user

Functional description

With the FTSHW command, you can display the appropriate file or files in a directory on a remote partner system.

There are three options for displaying attributes:
– List the name(s) of the file(s) in a directory
– Display a default selection of attributes returned by the partner system
– Display all attributes of a file or files in a directory, as returned by the partner system on request.

Format

FTSHW

PARTNER = <text 1..200 with-low>

,**FILE** = **_*NOT-SPECIFIED_** / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512> / ***DIR**ECTORY(...)

***DIR**ECTORY(...)
NAME = **_*NOT-SPECIFIED_** / <c-string 1..512 with-low> / <text 1..512> / <partial-filename 2..53>

,**PASS**WORD = **_*NONE_** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>

,**TRANS**FER-**ADM**ISSION = **_*NONE_** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
,**ACCOUNT** = **_*NONE_** / <c-string 1..64 with-low> / <text 1..64>
,**PASS**WORD = **_*NONE_** / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>

,**INF**ORMATION = **_*STD_** / ***ALL-ATTR**IBUTES / ***NAMES-ONLY**

,OUTPUT = **_*STDERR_**(...) / ***STDOUT**(...)

*STDERR(...) / ***STDOUT**(...)
LAYOUT = **_*STD_** / ***CSV**

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined by the FT administrator in the partner list or the address of the partner system. For more information on address specifications, see section “Defining the partner computer” on page 99.

FILE =
Name of the file in the remote FT partner system.

FILE = _*NOT-SPECIFIED_
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the file in the remote system. The file name must be specified in the syntax of the remote system and conform to the conventions of the remote system.
If the file name is specified with an unattached Public Volume Set, the request is rejected with the error message FTR2202.

FILE = *DIRECTORY(...)
Name of the directory.

NAME =
Name of the directory in the remote FT partner system.

NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

NAME = <c-string 1..512 with-low> / <text 1..512> / <partial-filename 2..53>
Name of the directory in the remote FT partner system. The directory name must be specified in the syntax of the remote system and must conform to the conventions of the remote system
If the remote system is a BS2000 system, you can specify a partially qualified file name, e.g. HUGO. All file names addressed by the partial qualification (e.g. HUGO.MAIER, HUGO.MULLER) are output.

If the partner is a BS2000 system and the file name is the name of a file generation group then the request is rejected with message FTR2148:
`Remote system: Transfer of file generation groups not supported`

PASSWORD =
Password that allows the user to access the file attributes in the remote system. If the file in the remote system is protected by a password, the password must be specified in the operands required to read file attributes in the remote system. If the remote system is a BS2000 or Unix system, no password is required.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password that allows the user to access the file in the remote system. The password must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or recognize any user authorization.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an FT profile if FTAC functionality has been installed on the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified. The alphanumeric entry is converted internally to lowercase characters.

TRANSFER-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..64 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

PASSWORD =
Password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD =
<c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and must be recognized by the remote system.

INFORMATION =
Amount of information required. The amount of information is also dependent on the amount of information supplied by the partner. Therefore, only some attributes may be displayed, even if you requested full information.

INFORMATION = *STD
The default range of information is output.

INFORMATION = *ALL-ATTRIBUTES
All available information is requested on the file in the partner system. However, only attribute values returned by the partner system can be displayed.

INFORMATION = *NAMES-ONLY
Only the names of the directory files or the name of the file is listed.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output is in **C**haracter **S**eparated **V**alues format. This is a special, tabular format, widely used in the PC world, in which the individual fields are separated by a semicolon ";" (see section "Output in CSV format" on page 164).

Examples

1. You want to view the properties of the PS file *dat1* with a variable record format of the length 255 on a z/OS system with the partner name *ZOS2PART*:

   To do this, you can issue the following command under TSO:

```
FTSHW ZOS2PART,FILE-NAME=DAT1,TRANS-ADM=(USERID,ACCT,PASSWORD),INF=*ALL

FILENAME=DAT1
 CRE   OPFT000
 MOD   DATE=Mar 25 00:00
 RECORD-FORMAT=v RECORD-SIZE=255        FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=8192
```

2. You want to view the properties of the file *dat2* on a BS2000 system with the partner name *BS2PART*:

```
FTSHW BS2PART,FILE-NAME=DAT2,TRANS-ADM=(USERID,ACCT,PASSWORD),INF=*ALL

FILENAME=DAT2
 CRE   OPENFT
 MOD   DATE=Mar 10 2012
 CHARACTERSET=g
 RECORD-FORMAT=f RECORD-SIZE=80
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=2048
```

3. You want to view the properties of the file *dat3* on the Windows system with the partner name *WINPC* and to do so, specify the FTAC transfer admission *FTACTRANSADM*:

```
FTSHW WINPC,FILE-NAME=DAT3,TRANS-ADM=('FTACTRANSADM')

*r----a----- FT:mueller    27185709  Sep 17 2011  DAT3
```

   The command for detailed output is as follows:

```
FTSHW WINPC,FILE-NAME=DAT3,TRANS-ADM=('FTACTRANSADM'),INF=*ALL

FILENAME=DAT3
 CRE   FT:mueller
 MOD   DATE=Sep 17 2011
 RECORD-FORMAT=u FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r----a-----   FILESIZE=27185709
```

4. You want to view the properties and members of the PO library *OPFT000.POBIB.CLIST* on a z/OS system with the user ID *opft000* and the partner name *ZOS2PART*:

```
FTSHW ZOS2PART,*DIR=('OPFT000.POBIB.CLIST')
              ,TRANS-ADM=(OPFT000,ACCT,PASSWORD)

 *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER1
 *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER2
 *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER3
```

The command for detailed output is as follows:

```
FTSHW ZOS2PART,*DIR=('OPFT000.POBIB.CLIST')
              ,TRANS-ADM=(OPFT000,ACCT,PASSWORD),INF=*ALL

 FILENAME=MEMBER1
 CRE   OPFT000
 MOD   DATE=Jul 16 00:00
 RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
 FILENAME=MEMBER2
 CRE   OPFT000
 MOD   DATE=Jul 16 00:00
 RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
 FILENAME=MEMBER3
 CRE   OPFT000
 MOD   DATE=Jul 16 00:00
 RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
```

5.21 FTSHWADS Display admission sets

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

You use the FTSHWADS command to display admission sets. You can output the following information on either SYSTSPRT or SYSPRINT:

– if the admission set is privileged (if so, then you are the FTAC administrator).

– if a password is required to use FTAC commands on this user ID. The password itself is not displayed.

– the limiting values for accessible security levels which have been set by the owner of this user ID.

– the limiting values for accessible security levels which have been pre-set by the FTAC administrator.

Format

FTSWHADS

USER-IDENTIFICATION = ***OWN** / ***ALL** / ***STD** / <name 1..8>

,**SEL**ECT-**PAR**AMETER = ***ALL**

,OUTPUT = ***STDERR**(...) / ***STDOUT**(...)

***STDERR**(...) / ***STDOUT**(...)
LAYOUT = ***STD** / ***CSV**

Operands

USER-IDENTIFICATION =
User ID whose admission set you wish to view. FTAC users can only obtain information about their own admission set and the default admission set. The FTAC administrator can obtain information about any admission set.

USER-IDENTIFICATION = *OWN
FTAC outputs your own user ID's admission set.

USER-IDENTIFICATION = *ALL
FTAC outputs the default admission set and the admission set of your own user ID.

USER-IDENTIFICATION = *STD
FTAC only outputs the default admission set.

USER-IDENTIFICATION = <name 1..8>
FTAC outputs the admission set that belong to the of the user ID specified. The FTAC user can only enter his/her own user ID here.

SELECT-PARAMETER = *ALL
This parameter is reserved for future extensions and has no effect in the current version.

OUTPUT =
Output medium for the information requested.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stderr.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stdout.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon ";" (see section "Output in CSV format" on page 164).

Example

Jack John, the FTAC administrator of the Dack Bank, wants to obtain information about the admission sets in his system. He enters the command

```
FTSHWADS␣USER-IDENTIFICATION=*ALL
```

Short form:

```
FTSHWADS␣*ALL
```

He receives the following output:

```
                MAX. USER LEVELS                MAX. ADM LEVELS           ATTR
USER-ID   OBS   OBR   IBS   IBR   IBP   IBF   OBS   OBR   IBS   IBR   IBP   IBF
*STD       10    10    10    10     0     0    10    10    10    10     0     0
JACK      100   100     0     0    0*    0*   100   100     0     0    0*    0* PRIV
GRACE      50    50   10*    50    50    50    50    50    50    50    50    50  PW
DANIEL      0    10     0     0     0     0    10    10     0     0     0     0  PW
STEVEN     50   100     0   10*     0     0    50   100    10    50     0     0
```

These can be explained as follows:

The user ID of each admission set is in the column USER-ID. In this example, there is a default admission set as well as admission sets for the user IDs JACK, GRACE, DANIEL and STEVEN.
The FTAC user sees the default admission set and his/her own admission set.

The column ATTR indicates the privileged admission set. We can see that JACK is the FTAC administrator.

The column ATTR also indicates whether an FTAC password has been defined (with PW). JACK, GRACE and DANIEL have done this to prevent others from using FTAC commands on their user ID which could be used to make modifications.

In the six columns under MAX-USER-LEVELS, the limiting values are output which the FTAC users have set for their admission sets. The six columns under MAX-ADM-LEVELS show the limiting values which the FTAC administrator has set. The smaller of the two values indicates up to which security level the owner of the admission set may use each basic function. The basic functions are abbreviated in the output as follows:

OBS = **O**UT**B**OUND-**S**END

OBR = **O**UT**B**OUND-**R**ECEIVE

IBS = **I**N**B**OUND-**S**END

IBR = **I**N**B**OUND-**R**ECEIVE

IBP = **I**N**B**OUND-**P**ROCESSING

IBF = **I**N**B**OUND-**F**ILEMANAGEMENT

The default admission set is configured such that it permits file transfers with systems which have the security level of 10 or lower, but does not permit any follow-up processing initiated by external sources (IBP=0). JACK may contact all available partner systems (OBS=100,OBR=100), but does not permit any file transfer accesses from outside onto his user ID (IBS=0,IBR=0,IBP=0).

The user ID GRACE is permitted to communicate with all partner systems with the security level of 50, according to the FTAC administrator's specifications. To better protect her files from strangers, GRACE has only made the function "inbound send" available to partner systems with the security level f 10 or lower.

The user ID DANIEL is heavily protected. Only files from partner systems with a maximum security level of 10 may be requested. A * after a number indicates that this value was taken from the default admission set and will change if any modifications are made to the default admission set.

5.22 FTSHWINS Display an openFT instance

Note on usage

User group: FT user

This command must be called under TSO.

The command only works if openFT has been started as a subsystem. If openFT has been started as a batch job, the message `INSTANCES MGMT NOT AVAILABLE` is issued.

Functional description

With the command FTSHWINS you can display information regarding openFT instances.

Format

FTSHWINS
[***ALL**]

Operands

INSTANCES = *ALL
Outputs a list of all available instances.

FTSHWINS with no operand specified displays the currently set instance.

Example

```
FTSHWINS *ALL
NAME   FTID   PORT   IP-ADDRESS          SERVICES    TIME    DATE
-------------------------------------------------------------------
STD    PBFT2  1100   123.45.67.89              44   7:51:06  2012-145
FTBLA  KUHEM  3040   123.45.67.99               3  15:16:33  2012-146
```

5.23 FTSHWLOG
Display log records and offline log files

Note on usage

User group: FT user, FT administrator and FTAC administrator

Functional description

With the FTSHWLOG command, you can obtain information on all FT requests logged by openFT. An important prerequisite is that the FT administrator has switched on the FT logging function. The logging records are marked as FT or FTAC or ADM, enabling you to identify the type of logging record.

FTSHWLOG also enables the name of the current log file and the names of the offline log files to be displayed.

FT logging

The FT user can view all log records which relate to his/her user ID. If no options are specified, openFT outputs the most recent log record. When requested, openFT outputs all the log records which correspond to the selection criterion defined in the command.

Command execution may take several minutes, depending on the size of the log file!

There are three types of output: short output and long output and CSV format.

FTAC logging

With FTAC functionality, FTSHWLOG can be used to display the FTAC log records. The FT user can view all FT log records, of which he/she is the owner.

If the access check was positive and openFT accepted the request, a second logging record is created in openFT, indicating whether the request was completed successfully, and if not, why it was terminated.

A precise description of output can be found starting on page 271.

Format

(part 1 of 2)

FTSHWLOG

SELECT = ***OWN** / ***ALL** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

LOGGING-ID = ***ALL** / <alphanum-name 1..12> / ***INTERVAL**(...)

***INTERVAL**(...)

FROM = **1** / <alphanum-name 1..12>

,TO = ***HIGHEST-EXISTING** / <alphanum-name 1..12>

,OWNER-IDENTIFICATION = ***OWN** / ***ALL** / <name 1..8>

,CREATION-TIME = ***INTERVAL**(...) / ***DAYS**(...)

***INTERVAL**(...)

FROM = **2000-01-01**(...) / <date 8..10>(...)

<date 8..10>(...)

TIME = **00:00** / <time 1..8>

,TO = ***TOMOR**ROW(...) / ***TODAY**(...) / <date 8..10>(...)

<date 8..10>(...)

TIME = **00:00** / <time 1..8>

***DAYS**(...)

NUMBER = <integer 1..1000>

,RECORD-TYPE = ***ALL** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

FT = ***TRANS**FER-**F**ILE / ***NONE** / list-poss(1): ***TRANS**FER-**F**ILE

,FTAC = (***TRANS**FER-**F**ILE, ***READ-FILE-ATTRIBUTES, *DEL**ETE-**FILE,**
***CRE**ATE-**FILE**, ***MOD**IFY-**FILE-ATTR**IBUTES,
***READ-DIRECTORY, *MOVE-FILE, *CREATE-DIRECTORY,**
***DELETE-DIRECTORY, *MODIFY-DIRECTORY. *LOGIN**) / ***NONE** /
list-poss(11): ***TRANS**FER-**F**ILE / ***READ-FILE-ATTR**IBUTES / ***DEL**ETE-**FILE** /
***CRE**ATE-**FILE** / ***MOD**IFY-**FILE-ATTR**IBUTES / ***READ-DIRECTORY** /
***MOVE-FILE** / ***CREATE-DIRECTORY** / ***DELETE-DIRECTORY** /
***MODIFY-DIRECTORY** / ***LOGIN**

,ADM = ***ADMIN**ISTRATION / ***NONE** / list-poss(1): ***ADMIN**ISTRATION

,INITIATOR = (***LOC**AL, ***REMOTE**) / list-poss(2): ***LOC**AL / ***REMOTE**

,PARTNER = ***ALL** / <text 1..200 with-low>

,FILE-NAME = ***ALL** / <filename 1..59> / <filename-prefix 2..50> / <c-string 1..512 with-low> /
***DIR**ECTORY(...)

***DIR**ECTORY(...)

NAME = ***ALL** / <partial-filename 2..50> / <c-string 1..512 with-low>

,REASON-CODE = ***ALL** / ***FAILURE** / <text 1..4>

(part 2 of 2)

```
        ,ROUTING-INFO = *ALL / <text 1..200 with-low>
        ,TRANSFER-ID = *ALL / <integer 1.. 2147483647>
        ,GLOBAL-REQUEST-ID = *ALL / <alphanum-name 1..10>
        ,LOGGING-FILE = *CURRENT / <filename 1..42> /  *ACTIVE-AT(...)
            *ACTIVE-AT(...)
                DATE = <date 8..10>
                ,TIME = 00:00 / <time 1..8>
        ,PREVIOUS-FILES = *STD / <integer 0..3>
,NUMBER = 1 / *ALL / <integer 1..99999999> / *POLLING(...)
    *POLLING(...)
        INTERVAL = 1 / <integer 1..600>
        ,NUMBER = *UNLIMITED / <integer 1..3600>
,INFORMATION = *STD / *ALL / *LOGGING-FILES
,OUTPUT = *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
        LAYOUT = *STD / *CSV
```

Operands

SELECT =
Selects a group of logging records.

SELECT = *OWN
Selects logging records under the user's own login.

SELECT = *ALL
As user you receive only logging records of your own ID (same as *OWN).

SELECT = *PARAMETERS(...)

LOGGING-ID =
Number of the logging record.

LOGGING-ID = *ALL
The number of the logging record is not a selection criterion.

LOGGING-ID = <alphanum-name 1..12>
Number of the logging record to be output. The value range for the logging ID is from 1 through 999999999999.

LOGGING-ID = *INTERVAL(...)
Range of logging records to be output.

FROM = <alphanum-name 1..12>
First logging record to be output. The value range for the logging ID is from 1 through 999999999999.

TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
Last logging record to be output. The value range for the logging ID is from 1 through 999999999999.

OWNER-IDENTIFICATION =
User ID whose logging records are to be displayed.

OWNER-IDENTIFICATION = *OWN
Logging records of your user ID are displayed.

OWNER-IDENTIFICATION = *ALL
Normal FT users receive information only on the logging records of their own respective IDs even if *ALL is specified.

OWNER-IDENTIFICATION = <name 1..8>
FT users may only specify their own ID.

CREATION-TIME =
The range of the logging records to be output, selected by their date or time of creation.

CREATION-TIME = *INTERVAL(...)
The range is specified as a time interval using the date and/or time.

FROM = 2000-01-01(...) / <date 8..10>(...)
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 20012-08-18 or 12-08-18 for 18 August, 2012. openFT then displays all logging records written after the specified date and time.

TIME = 00:00 / <time 1..8>
Time for the day specified with CREATION-TIME. openFT displays all logging records written after the specified time. The time is entered in the format *hh:mm:ss*, e.g. 14:30:10.

TO = *TOMORROW / *TODAY(...) / <date 8..10>(...)
Creation date up to which the log records are to be displayed.

TO = *TOMORROW
Outputs all log records which were created by the time of the command output.

TO = *TODAY
When CREATION-TIME is used to explicitly specify a time, all log records which were written up to this time are displayed. If no time was specified, openFT displays all log records which were written up to and including at midnight on the previous day.

TO=<date 8..10>(...)
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 20012-08-18 or 12-08-18 for 18 August, 2012. openFT then displays all logging records up to the specified time.

TIME = 00:00 / <time 1..8>
Time for the day specified with CREATION-TIME. openFT displays all logging records written up to the specified time. The time is entered in the format *hh:mm:ss*, e.g. 14:30:10.

CREATION-TIME = *DAYS(NUMBER=<integer 1..1000>)
This field is specified in number of days. All logging sets that were created in the last n calendar days, including today, are output.

RECORD-TYPE =
Type of logging record to be displayed.

RECORD-TYPE = *ALL
The record type is not a selection criterion.

RECORD-TYPE = *PARAMETERS(...)
Type of the logging record.

FT = *TRANSFER-FILE / *NONE / list-poss(1): *TRANSFER-FILE
Specifies whether or not the FT logging records are to be displayed.

FTAC =
(*TRANSFER-FILE, *READ-FILE-ATTRIBUTES, *DELETE-FILE, *CREATE-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *MOVE-FILE, *CREATE-DIRECTORY, *DELETE-DIRECTORY, *MODIFY-DIRECTORY, *LOGIN) / *NONE / list-poss(11): *TRANSFER-FILE / *READ-FILE-ATTRIBUTES / *DELETE-FILE / *CREATE-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY / *MOVE-FILE / *CREATE-DIRECTORY / *MODIFY-DIRECTORY / *DELETE-DIRECTORY / *LOGIN
Specifies whether or not FTAC logging records are to be displayed. If they are to be displayed, the FT function for which the FTAC logging records are to be displayed can also be specified. The following values are possible:

*TRANSFER-FILE
All logging records for the function “Transfer files” are displayed.

*READ-FILE-ATTRIBUTES
All logging records for the function “Read file attributes” are displayed.

*DELETE-FILE
All logging records for the function “Delete files” are displayed.

*CREATE-FILE
All logging records for the function “Create files” are displayed.

*MODIFY-FILE-ATTRIBUTES
All logging records for the function “Modify file attributes” are displayed.

*READ-DIRECTORY
All logging records for the function “Read file directory” are displayed.

*MOVE-FILE
All logging records for the function “Copy and delete files” are displayed.

*CREATE-DIRECTORY
All logging records for the function “Create directory” are displayed.

*DELETE-DIRECTORY
All logging records for the function “Delete directory” are displayed.

*MODIFY-DIRECTORY
All logging records for the function “Modify directory” are displayed.

*LOGIN
All logging records for the function “Inbound FTP access” are displayed. Log records of the type *LOGIN are only written in the case of an incorrect transfer admission.

ADM = *ADMINISTRATION / *NONE / list-poss(1): *ADMINISTRATION
Specifies whether ADM log records are output.

ADM = *ADMINISTRATION
ADM log records are output. For further details, refer to the openFT manual "Installation and Administration".

ADM = *NONE
No ADM log records are output.

INITIATOR =
Logging records according to the initiator.

INITIATOR = (*LOCAL,*REMOTE)
The initiator is not a selection criterion.

INITIATOR = *LOCAL
Only those logging records that belong to requests issued locally are displayed.

INITIATOR = *REMOTE
Only those logging records belonging to requests made from a remote system are displayed.

PARTNER =
The partner system.

PARTNER = *ALL
The partner system is not a selection criterion.

PARTNER = <text 1..200 with-low>
Name or address of the partner system for which the logging records are to be displayed. For more information on address specifications, see section “Defining the partner computer” on page 99.

For the partner name, you can also use the wildcard symbols '*' (asterisk) and '?' (question mark). ’*’ stands for any string and ’?’ stands for any single character. The asterisk may not, however, be in first place. You can enter '?*' instead.

FILE-NAME =
File name.

FILE-NAME = *ALL
The file name is not a selection criterion.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Fully qualified name of the files for which you wish to view the logging records.

FILE-NAME = <filename-prefix 2..50>
Partially qualified name of the files for which you want to view the logging records.

Examples
- If you specify TOOLS as the beginning of the filename, all logging records containing the filename TOOLS.CLIST, TOOLS.CNTL or TOOLS.CLIST(MEMBER01) will be displayed.
- If you specify TOOLS.CLIST/ as the beginning of the filename, all logging records containing the filename TOOLS.CLIST(MEMBER01), TOOLS.CLIST(MEMBER02), etc. are displayed.

FILE-NAME = *DIRECTORY(...)
Name of the directory.

***DIRECTORY(...)**
Here you specify the directory in the same format as used on the partner computer in one of the openFT user commands CREATE-/MODIFY-/DELETE-REMOTE-DIR or FTSHW (see page 248).

NAME = *ALL
The directory is not a selection criterion

NAME = <partial-filename 2..50> / <c-string 1..512 with-low>
Name of the directory.

Example
If you specify FILE=*DIR(NAME=ABC.) here, and not FILE=ABC., only those logging records are displayed that contain ABC (as the name of a file directory which were accessed from a remote system with the file management command in order to display an z/OS file directory).

REASON-CODE =
Selection by the reason code of the logging records.

REASON-CODE = *ALL
The reason code is not a selection criterion; all records are output.

REASON-CODE = *FAILURE
All logging records with error codes are output.

REASON-CODE = <text 1..4>
Logging records to be output by the error codes. Leading zeros can be omitted (e.g. 14 for FTR0014).

ROUTING-INFO = *ALL / <text 1..200 with-low>
Selects the ADM log records on the basis of the routing information. The routing information describes the administered instance in the case of remote administration requests issued locally.

ROUTING-INFO = *ALL
The routing information is not used as a selection criterion.

ROUTING-INFO = <text 1..200 with-low>
Routing information for which the ADM log records are to be output.

TRANSFER-ID =
Selection on the basis of the request ID.

TRANSFER-ID = *ALL
The request ID is not used as a selection criterion.

TRANSFER-ID = <integer 1..2147483647>
Only outputs log records for the specified request ID.

GLOBAL-REQUEST-ID = *ALL / <alphanum-name 1..10>
Selects the log records on the basis of the global request ID.

GLOBAL-REQUEST-ID = *ALL
The global request identification is not a search criterion.

GLOBAL-REQUEST-ID = <alphanum-name 1..10>
Outputs log records for the specified global request identification. The global request identification is relevant only for inbound requests of openFTpartners. It is assigned by the initiator of the request (transfer ID) and transferred to the local system.

LOGGING-FILE =
Selects the log file whose logging records or name are to be output. This means that you can also view offline log records.

LOGGING-FILE = *CURRENT
The current log file is selected.

LOGGING-FILE = <filename 1..42>
Specifies the name of the log file which is to be searched. If you specify a value > 0 in the PREVIOUS-FILES operand, further, older offline log files are also searched (if any exist).

LOGGING-FILE = *ACTIVE-AT(...)
Selects the log file using its creation time (local time). The log file which was created on or before the specified time is selected. If more than one log file matches the specified time, the most recent of these log files is selected. If you specify a value > 0 in the PREVIOUS-FILES operand, further, older offline log files are also searched (if any exist).

DATE = <date 8..10>
Creation date in the format *yyyy-mm-dd* or *yy-mm-dd*, z.B. 2012-01-31 or 12-01-31 for Januray 31, 2012.

TIME = 00:00 / <time 1..8>
Creation time on the date specified with DATE. You specify the time in the format *hh:mm:ss*, e,.g. 14:30:10.

PREVIOUS-FILES =
Specifies the number of preceding offline log files that are to be selected in addition to the current file or the file specified with LOGGING-FILE.

PREVIOUS-FILES = *STD
The effect depends on the specification in the INFORMATION operand:

– INFORMATION = *STD (default value) or *ALL: The current lo file or the log file specified with LOGGING-FILE is searched for log records.

– INFORMATION = *LOGGING-FILES: The names of all log files are output (maximum of 1024).

PREVIOUS-FILES = <0..3>
Specifies the number of preceding offline log files (0 to 3) that are to be searched in addition to the current file or the file specified with LOGGING-FILE or whose names are to be output.

NUMBER =
Maximum number of log records or polling intervals for outputting log records.

NUMBER = 1 / <integer 1..99999999>
The maximum number of logging records that are to be displayed. The default value is 1.

NUMBER = *ALL
All logging records are displayed.

NUMBER = *POLLING(...)
Specifies that the output of log records will be repeated at regular intervals. You can define the polling interval and the number of repetitions. Irrespective of the specifications in INTERVAL and NUMBER, the most recent log record which exists is always output first.

INTERVAL = 1 / <integer 1...600>
Polling interval in seconds. On each repetition, all the new log records are filtered in accordance with the specified selection criteria and the detected records are output. By default the output is repeated every second.

NUMBER =
Number of repetitions.

NUMBER = *UNLIMITED
The output is repeated without restriction. You can, for example, cancel the output using the key combination PA1 and RESET.

NUMBER = <integer 1..3600>
Specifies the number of reperitions.

NUMBER = *POLLING may not be combined with the following specifications:
- LOGGING-FILE = <filename ..>
- LOGGING-FILE = *ACTIVE-AT(...)
- INFORMATION = *LOGGING-FILES
- TRANSFER-ID = <integer 1..2147483647>
- GLOBAL-REQUEST-ID = <alphanum-name 1..10>
- LOGGING-ID = <alphanum-name 1..12> / *INTERVAL(...)
- CREATION-TIME = *INTERVAL(...) / *DAYS(...)
- PREVIOUS-FILES = <integer 0..3>

INFORMATION =
Scope of the requested information.

INFORMATION = *STD
The logging records are displayed in a standard format (see page 269).

INFORMATION = *ALL
The logging records are displayed in a detailed format (see page 271).

INFORMATION = *LOGGING-FILES
Outputs only the names of the log file(s).

INFORMATION = *LOGGING-FILES can only be combined with the following parameters:
- LOGGING-FILE in SELECT=*PARAMETERS(…)
- PREIOUS-FILES in SELECT=*PARAMETERS(…)
- OUTPUT

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon ";" (see page 164).

5.23.1 Description of the short output

Short output form of FT logging records (example)

```
FTSHWLOG NUMBER=2

 TYP LOGG-ID  TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2012-04-22
 T       5333 14:18:24 2169 <G133H301 FT2V292        FT2V292  TEST2
 T       5284 14:08:12 0000 >G133H301 FT2V292        FT2V292  TEST1

FTSHWLOG NUMBER=2

 TYP LOGG-ID  TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2012-04-22
 T       5333 14:18:24 2169 <G133H301 FT2V292        FT2V292  TEST2
 T       5284 14:08:12 0000 >G133H301 FT2V292        FT2V292  TEST1
```

Explanation

Not all values are displayed for all log record types and request types.

The table below also describes values that can occur only in ADM log records.

Name	Explanation	
TYP (column 1)	Specifies if it is an FT or FTAC or ADM or FTP log record. T indicates the FT logging record, C indicates the FTAC logging record, A indicates the ADM logging record.	
TYP (columns 2-3)	Definition of FT function:	
	␣	transfer file
	V	transfer file and delete send file (only inbound possible)
	A	read file attributes
	D	delete file
	C	create file
	M	modify file attributes
	R	read directory
	CD	create director
	MD	modify directory
	DD	delete directory
	L	login (inbound FTP access)
LOGG-ID	Number of the log record (up to twelve digits)	
TIME	Time when the logging record was written	
RC	Reason Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for "FTAC reasons" (e.g. 0014), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.	
PARTNER	Provides information about the partner system. The output in the case of named partners consists of the symbolic name, and in the case of dynamic partners of the address (up to 8 characters; if the address is longer, the last character is an'*'). The partner system is prefixed by an identifier from which you can determine the request direction.	
	>	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner – request to view remote file attributes – request to view remote directories
	<	The request direction is to the local system. This direction is specified for a – receive request, i.e.the data is transferred to the local system – request to modify remote file attributes – request to delete remote files

Name	Explanation
INITIATOR	Initiator (user ID) in the case of requests issued locally issued; if initiative is from remote system: *REMOTE
INIT	The field is always empty in z/OS and is only output for reasons of compatibility.
USER-ADM	User ID in the local system used by the requests
FILENAME	Filename resp. pre-processing or post-processing in the local system. In the case of ADM logging records, this field is empty. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV pre-processing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.

5.23.2 Description of the long output

Long output form outbound (example)

```
LOGGING-ID  = 9479      RC      = 0000        TIME     = 2012-07-11 14:31:29
   TRANS    = TO        REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-07-11 14:31:29
   TRANS-ID = 67052     WRITE   = REPLACE     REQUESTED= 2012-07-11 14:31:28
   TRANSFER =         1 kB                    CCS-NAME = IBM1047
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= OPFTUID
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = FILE.TEST

LOGGING-ID  = 9478      RC      = 0000        TIME     = 2012-07-11 14:31:28
   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  =           PRIV    =
   INITIATOR= OPFTUID
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = FILE.TEST
```

Long output form inbound (example)

```
LOGGING-ID  = 9473      RC      = 0000        TIME     = 2012-07-11 14:25:00
   TRANS    = FROM      REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-07-11 14:24:59
   TRANS-ID = 67046     WRITE   = REPLACE     STORETIME= 2012-07-11 14:25:00
   TRANSFER =         1 kB                    CCS-NAME = IBM1047
   SEC-OPTS = ENCR+DICHK+DENCR+DDICHK+LAUTH2+RAUTH2, RSA-1024 / AES-256
   INITIATOR= *REMOTE                         GLOB-ID  = 66279
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = TEST1

LOGGING-ID  = 9472      RC      = 0000        TIME     = 2012-07-11 14:24:59
   TRANS    = FROM      REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  = PROFIL1   PRIV    = NO
   INITIATOR= *REMOTE                         GLOB-ID  = 66279
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = TEST1
```

Explanation of long output form (column-wise)

The table below also describes fields and values that can only occur in ADM log records.

Name	Explanation	
LOGGING-ID	Number of the log record (up to twelve digits)	
TRANS	Transfer direction:	
	TO	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner. – request to view remote file attributes – request to view remote directories
	FROM	The request direction is to the local system (inbound). This direction is specified for a – receive request, i.e. the data are transferred to the local system – request to modify remote file attributes – request to delete remote files
	BOTH	File management request with two-way data transfer.
PROFILE	Name of the profile to be used for the transfer (empty in the FT logging record)	
TRANS-ID	Transfer ID number	
TRANSFER	Amount of data transferred	

Name	Explanation	
SEC-OPTS	Security options and encryption algorithms used. This line is only output if at least one of the options is used.	
	ENCR	Encryption of the request queue
	DICHK	Data integrity check of the request queue
	DENCR	Encryption of data content during the transfer
	DDICHK	Data integrity check of the file data to be transferred
	LAUTH	Authentication of the local system on a partner (authentication level 1)
	LAUTH2	Authentication of the local system on a partner (autehtication level 2)
	RAUTH	Authentication of the partner on a local system (authentication level 1)
	RAUTH2	Authentication of the partner on a local system (authentication level 2)
	RSA-nnnn	Length of the RSA key
	DES / AES-128 / AES-256	Encryption algorithm used
INITIATOR	Initiator (user ID) in the case of requests issued locally issued; if initiative is from remote system: *REMOTE	
USER-ADM	User ID in the local system used by the requests	
PARTNER	Provides information about the partner system. The output includes the symbolic name under which the system administrator has entered the partner system in the partner list. If dynamic partners are admitted, the partner system can be output as partner address.	
FILENAME	Filename resp. pre-processing or post-processing in local system. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV pre-processing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.	
ADM-CMD	Only output for an ADM log record: Administration command without parameters	
ADMIN-ID	Only output for an ADM log record: Remains always empty in z/OS because only relevant on the remote adminsitration server	
ROUTING	Only output for an ADM log record:: Routing information on the openFT instance to be administered	

Name	Explanation	
RC	Reason-Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for “FTAC reasons” (e.g. 2169), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.	
REC-TYPE	Specifies if this is an FT or FTAC or ADM logging record.	
PCMD	Status of follow-up processing:	
	NONE	No follow-up processing defined.
	STARTED	Follow-up processing was started.
	NOT-STARTED	Follow-up could not be started.
PRIV	specifies whether the admission profile is privileged.	
WRITE	Write rules:	
	NEW	A new file is created. If a file with the same name already exists, the transfer will be aborted.
	EXT	An existing file is extended and stored as new.
	REPLACE	An existing file is extended.
TIME	Time when the logging record was written	
FUNCTION	Definition of FT function:	
	– TRANSFER-FILE: transfer file – MOVE-FILE: transfer file and delete send file (only inbound possible) – READ-FILE-ATTRIBUTES: read file attributes – DELETE-FILE: delete file – CREATE-FILE: create new file – MODIFY-FILE-ATTRIBUTES: modify file attributes – READ-DIRECTORY: read directory – CREATE-DIRECTORY: create directory – MODIFY-DIRECTORY: modify directory – DELETE-DIRECTORY: delete directory – LOGIN: inbound FTP access – REM-ADMIN: remote administrator	
STARTTIME	Time request was started	
STORETIME	Time request was accepted (inbound)	
REQUESTED	Time request was accepted (outbound)	
CCS-NAME	Name of the character set, used for code conversion as necessary.	
CHG-DATE	Specifies whether the change date of the send file is taken over for the receive file.	
	SAME	The change date of the send file is take over.
INITSN	TSN from which the request came, entered only in the case of outbound requests.	

Name	Explanation
GLOB-ID	Global request identification, displayed only in the case of inbound requests from openFT and FTAM partners (INITIATOR=REMOTE). This corresponds to the request identification (=TRANSFER-ID) on the initiator system.

Example

The FT administrator wants to display all logging records that were created for the user ID *Meier* and logged between 01.01.2012 and 31.03.2012. If you are the owner of the User ID *Meier*, you can omit the parameter OWNER-IDENTIFICATION=.

```
FTSHWLOG SELECT=*PARAMETERS(OWNER-IDENTIFICATION=MEIER,          -
                 CREATION-TIME=*INTERVAL(FROM=2012-01-01(00:00), -
                 TO=2012-03-31(23:59))),NUMBER=*ALL
```

You want to see the first record of the output in detail.

```
FTSHWLOG (OWN=Meier,CRE-TIME=*INTERVAL(FROM=2012-01-01(00:00),  -
          TO=2012-03-31(00:00))),INF=*ALL
```

5.24 FTSHWMON Show monitoring data

Note on usage

User group: FT users and FT administrators

Description of the function

The FTSHWMON command allows you to output the monitoring values from openFT operation on the local system. To do this, monitoring must be activated (see FTMODOPT in the System Administrator Guide) and openFT must be activated.

Format

```
FTSHWMON
 NAME = *STD / *ALL /<list-poss(100): alphanum-name 1..12>
,POLLING =*NONE / *PARAMETERS(...)
    *PARAMETERS(...)
        INTERVAL=1 /<integer 1..600>
        ,NUMBER=*UNLIMITED / <integer 1..3600>
,INFORMATION=*VALUES(...) / *TYPE
    *VALUES(...)
        DATA=*FORMATTED / *RAW
,OUTPUT= *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
        LAYOUT = *STD / *CSV
```

Operands

NAME =
Specifies what monitoring values are to be output.

NAME = *STD
A predefined default set of monitoring values is output, see “Examples” on page 284.

NAME = *ALL
All monitoring values are output.

NAME = <list-poss(100): alphanum-name 1..12>
Here you can enter a list of up to 100 names of monitoring values that are to be output. The name must be one of the short names (see the table in the section “Description of the monitoring values” on page 279).

POLLING =
Specifies the interval at which the monitoring values are to be polled.

POLLING =*NONE
The monitoring values are only polled once.

POLLING =*PARAMETERS
In this structure you specify a time interval and a repetition factor for polling the monitoring values. If an error occurs during polling, further repeated output is canceled.

INTERVAL = 1
The time interval for polling the monitoring values is 1 second.

INTERVAL = <integer 1..600>
Time interval in seconds for polling the monitoring values.

NUMBER = *UNLIMITED
There is no limit to the number of times the monitoring values are polled. To cancel the command, you can use the key combination PA1 and RESET, for example.

NUMBER = <integer 1..3600>
Here you specify how often the monitoring values are to be polled.

INFORMATION =
Specifies whether the monitoring values themselves or the type of the monitoring values is to be output.

INFORMATION = *VALUES(...)
The measured value is output. You can specify whether the monitoring values are to be output in formatted form or as raw data.

DATA =*FORMATTED
The monitoring values are formatted for visual display, e.g. as throughput, maximum or average.

DATA =*RAW
Raw, unformatted data is output. Monitoring values for the duration of an action are not output.

INFORMATION = *TYPE
Outputs the type and, where applicable, the scaling factor of the monitoring value or the type of the metadata.

The scaling factor is only of significance for some monitoring values and in CSV format if *RAW is not specified. In this case, the output value must be divided by the scaling factor to get the real value. In the case of formatted data in tabular format, the scaling factor 100 specifies that the number is output to 2 decimal places.

The following output values are possible for *TYPE:

*BOOL	Boolean value
*PERCENT	Percentage
*INT	Integer number (corresponds to *INT(1))
*INT(100)	Integer value with a scaling factor of 100
*TIME	Timestamp
*STRING	Text output for the selection

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
The data is output to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
The data is output to SYSPRINT.

LAYOUT = *STD
Output is formatted in a form readable by the user.
If the monitoring configuration changes (filters), a new header and a new start time for monitoring is output in standard output format.

LAYOUT = *CSV
Data is output in Character Separated Values format. This is a quasi-tabular format that is in widespread use in the field of PCs and in which the individual fields are separated by semicolons ";" (see section “Output in CSV format” on page 164).

If the monitoring configuration changes (filters), the new start time for monitoring is shown in a separate column in CSV format.

5.24.1 Description of the monitoring values

The table below shows all the monitoring values output when NAME=*ALL is specified. Under NAME=, you can also specify a list of any of the parameters shown in the table.

The first two letters of the name indicate the data object that the monitoring value belongs to.

– Th = Throughput
– Du = Duration
– St = State

The second component of the name indicates the performance indicator, e.g. Netb for net bytes. In the case of monitoring values for the Throughput or Duration data object, the last 3 letters of the name indicate the types of requests from which the monitoring value originates, e.g.

– Ttl = FT Total
– Snd = FT Send requests
– Rcv = FT Receive requests
– Txt = Transfer of text files
– Bin = Transfer of binary files
– Out = FT Outbound
– Inb = FT Inbound

If monitoring is deactivated for all partners (PARTNER-SELECTION=*NONE with FTMODOPT ...,MONITORING), only the following values are provided:

Status: StCLim, StCAct, StRqLim, StRqAct, StOftr, StFtmr, StFtpr, StTrcr

All the other values are set to 0.

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
ThNetbTtl	Throughput in net bytes: Number of bytes transferred	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbSnd	Throughput in net bytes (send requests): Number of bytes transferred with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbRcv	Throughput in net bytes (receive requests): Number of bytes transferred with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbTxt	Throughput in net bytes (text files): Number of bytes transferred when transferring text files	*ALL	Number of bytes per second	Bytes, accumulated

Name	Meaning	Output with	Output unit	
			FORMATTED	**RAW**
ThNetbBin	Throughput in net bytes (binary files): Number of bytes transferred when transferring binary files	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskTtl	Throughput in disk bytes: Number of bytes read from files or written to files with transfer requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskSnd	Throughput in disk bytes (send requests): Number of bytes read from files with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskRcv	Throughput in disk bytes (receive requests): Number of bytes written to files with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskTxt	Throughput in disk bytes (text files): Number of bytes read from text files or written to text files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskBin	Throughput in disk bytes (binary files): Number of bytes read from binary files or written to binary files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThRqto	openFT requests: Number of openFT requests received	*STD/ *ALL	Number per second	Accumulated number
ThRqft	File transfer requests: Number of file transfer requests received	*ALL	Number per second	Accumulated number
ThRqfm	File management requests: Number of file management requests received	*ALL	Number per second	Accumulated number
ThSuct	Successful requests: Number of successfully completed openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThAbrt	Aborted requests: Number of aborted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThIntr	Interrupted requests: Number of interrupted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThUsrf	Requests from non-authorized users: Number of openFT requests in which the user check was terminated with errors	*STD/ *ALL	Number per second	Accumulated number

Name	Meaning	Output with	Output unit	
			FORMATTED	**RAW**
ThFoll	Started follow-up processing operations: Number of follow-up processing operations started	*ALL	Number per second	Accumulated number
ThCosu	Connections established: Number of connections successfully established	*ALL	Number per second	Accumulated number
ThCofl	Failed connection attempts: Number of attempts to establish a connection that failed with errors	*STD/ *ALL	Number per second	Accumulated number
ThCobr	Disconnections: Number of disconnections as a result of connection errors	*STD/ *ALL	Number per second	Accumulated number
DuRqtlOut	Maximum outbound request duration: Maximum request duration of an outbound request	*ALL	Milliseconds [1]	-
DuRqtlInb	Maximum inbound request duration: Maximum request duration of an inbound request	*ALL	Milliseconds [1]	-
DuRqftOut	Maximum outbound transfer request duration: Maximum duration of an outbound file transfer request	*ALL	Milliseconds [1]	-
DuRqftInb	Maximum inbound transfer request duration: Maximum duration of an inbound file transfer request	*ALL	Milliseconds [1]	-
DuRqfmOut	Maximum outbound file management request duration: Maximum duration of an outbound file management request	*ALL	Milliseconds [1]	-
DuRqfmInb	Maximum inbound file management request duration: Maximum duration of an inbound file management request	*ALL	Milliseconds [1]	-
DuRqesOut	Maximum outbound request waiting time: Maximum waiting time before an outbound request is processed (for requests without a specific start time)	*ALL	Milliseconds [1]	-

Name	Meaning	Output with	Output unit	
			FORMATTED	**RAW**
DuDnscOut	Maximum duration of an outbound DNS request Maximum time an outbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuDnscInb	Maximum duration of an inbound DNS request Maximum time an inbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuConnOut	Maximum duration of establishment of a connection: Maximum time between requesting a connection and receiving confirmation of a connection for an outbound openFT request	*ALL	Milliseconds [1]	-
DuOpenOut	Maximum file open time (outbound): Maximum time an outbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuOpenInb	Maximum file open time (inbound): Maximum time an inbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuClosOut	Maximum file close time (outbound): Maximum time an outbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuClosInb	Maximum file close time (inbound): Maximum time an inbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuUsrcOut	Maximum user check time (outbound): Maximum time an outbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
DuUsrcInb	Maximum user check time (inbound): Maximum time an inbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
StRqas	Number of synchronous requests in the ACTIVE state	*STD/ *ALL	Average [2]	Current number
StRqaa	Number of asynchronous requests in the ACTIVE state	*STD/ *ALL	Average value [2]	Current number
StRqwt	Number of requests in the WAIT state	*STD/ *ALL	Average value [2]	Current number

Name	Meaning	Output with	Output unit	
			FORMATTED	**RAW**
StRqhd	Number of requests in the HOLD state	*STD/ *ALL	Average value [2]	Current number
StRqsp	Number of requests in the SUSPEND state	*STD/ *ALL	Average value [2]	Current number
StRqlk	Number of requests in the LOCKED state	*STD/ *ALL	Average value [2]	Current number
StRqfi	Number of requests in the FINISHED state	*ALL	Average value [2]	Current number
StCLim	Maximum number of connections: Upper limit for the number of connections established for asynchronous requests.	*STD/ *ALL	Value currently set	
StCAct	Number of occupied connections for asynchronous requests	*STD/ *ALL	Share of StCLim in % [3]	Current number
StRqLim	Maximum number of requests: Maximum number of asynchronous requests in request management	*STD/ *ALL	Value currently set	
StRqAct	Entries occupied in request management	*STD/ *ALL	Share of StRqLim in % [3]	Current number
StOftr	openFT protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtmr	FTAM protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtpr	FTP protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StTrcr	Trace activated/deactivated	*ALL	ON (activated) OFF (deactivated)	

[1] Maximum value during the last monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The minimum time interval output is 1 millisecond if a relevant measurement has been completed during the interval since the last query. A value of 0 specifies that no measurement has been made in this interval.

[2] Average value during the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The format is n.mm, where n is an integer and mm are to be interpreted as decimal places.

[3] If the reference value is reduced in live operation, it is possible for the value output to lie above 100 (%) temporarily.

5.24.2 Examples

1. Monitoring values are to be output in default output format.

```
FTSHWMON
openFT(STD)   Monitoring (formatted)
MonOn=2012-02-17 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2012-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl 38728
 ThNetbSnd 38728
 ThNetbRcv 0
 ThDiskTtl 16384
 ThDiskSnd 16384
 ThDiskRcv 0
 ThRqto    1
 ThSuct    0
 ThAbrt    0
 ThIntr    0
 ThUsrf    0
 ThCofl    0
 ThCobr    0
 StRqas    0.00
 StRqaa    8.66
 StRqwt    1.66
 StRqhd    0.00
 StRqsp    0.00
 StRqlk    0.00
 StCLim    16
 StCAct    37
 StRqLim   1000
 StRqAct   1
 StOftr    ON
 StFtmr    OFF
 StFtpr    OFF
```

Explanation

The default output format begins with a header containing the following specifications:
- Name of the openFT instance and selected data format (raw or formatted)
- Monitoring start time and partner and request selection
- Current timestamp

This is followed by the list of default values. See the section “Description of the monitoring values” on page 279 for the meanings.

2. Only the data types are to be output in default output format.

```
FTSHWMON INFORMATION=*TYPE
openFT(STD)   Monitoring (formatted)
MonOn=2012-02-17 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2012-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl INT
 ThNetbSnd INT
 ThNetbRcv INT
 ThDiskTtl INT
 ThDiskSnd INT
 ThDiskRcv INT
 ThRqto    INT
 ThSuct    INT
 ThAbrt    INT
 ThIntr    INT
 ThUsrf    INT
 ThCofl    INT
 ThCobr    INT
 StRqas    INT(100)
 StRqaa    INT(100)
 StRqwt    INT(100)
 StRqhd    INT(100)
 StRqsp    INT(100)
 StRqlk    INT(100)
 StCLim    INT
 StCAct    PERCENT
 StRqLim   INT
 StRqAct   PERCENT
 StOftr    BOOL
 StFtmr    BOOL
 StFtpr    BOOL
```

Explanation

The types in the Value column have the following significance:

INT	Integer number (corresponds to INT(1))
INT(100)	Numeric value with a scaling value of 100 in the format n.mm, where n is an integer and mm are decimal places.
PERCENT	Percentage
BOOL	Boolean value, ON / OFF

3. The monitoring value "throughput in netbytes" (ThNetbTtl) is to be displayed. The display is to be updated every 60 seconds and repeated three times (polling).

```
FTSHWMON NAME=ThNetbTtl,POLLING=*PAR(INTERVAL=60,NUMBER=3)

openFT(STD)   Monitoring (formatted)
MonOn=2012-02-19 10:44:09 PartnerSel=OPENFT,FTP  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2012-02-19 12:45:33
 Name      Value
 ----------------
 ThNetbTtl 780107

2012-02-19 12:46:33
 ThNetbTtl 993051

2012-02-19 12:47:33
 ThNetbTtl 1049832
```

The repetitions are separated by intermediate header containing the current polling time.

5.25 FTSHWOPT Display operating parameters

Note on usage

User group: FT user and FT administrator

Functional description

The command FTSHWOPT can be used at any time to obtain the information listed below on the operating parameters of your FT system:

- Information on whether or not openFT has been started
- Instance identification
- Maximum values for operation (maximum number of file transfer requests in the request file, maximum lifetime of requests, maximum number of processes and transport connections, maximum size of a transport unit)
- Security settings (FTAC security level of the partner systems, extended sender verification)
- Logging settings (scope, intervals for automatic deletion)
- Trace settings
- Settings for traps (console traps, ADM traps)
- Settings for the monitoring functions

Format

FTSHWOPT
OUTPUT = ***STDERR**(...) / ***STDOUT**(...) ***STDERR**(...) / ***STDOUT**(...) **LAYOUT** = ***STD** / ***CSV** / ***BS2-PROC** / ***ZOS-PROC**

Operands

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output takes place in **C**haracter **S**eparated **V**alues format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by semicolons ";" (see section "Output in CSV format" on page 164).

LAYOUT = *BS2-PROC
The operating parameters are output as a command sequence. This can be called as an SDF procedure at BS2000/OSD systems in order to recreate the identical operating parameters.

LAYOUT = *ZOS-PROC
The operating parameters are output as a command sequence. This can be called as a Clist procedure at z/OS systems in order to recreate the identical operating parameters.

5.25.1 Description of the output

Example

Default of the FTSHWOPT command, i.e. the operating parameters have not been modified since installation.

```
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE KEY-LEN CCS-NAME
  YES       2       16        8     2000      30       65535   2048   IBM1047
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG     ENC-MAND
  STD       ON   B-P-ATTR    ALL    ALL    ALL           NO
OPENFT-APPL      FTAM-APPL          FTP-PORT           ADM-PORT
1100             *NONE              21                 11000
ACTIVE           NAVAIL             ACTIVE             ACTIVE
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
MCHZPDT2         FJMPBFT2 / $FJAM,FJMPBFT2

DEL-LOG  ON  AT     RETPD    ADM-TRAP-SERVER
 OFF   DAILY 00:00    14     *NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS    OFF      OFF       OFF        OFF       OFF       OFF        OFF
ADM              OFF       OFF        OFF       OFF       OFF        OFF

FUNCT:  SWITCH PARTNER-SELECTION    REQUEST-SELECTION    OPTIONS
MONITOR  OFF   ALL                  ALL
TRACE    OFF   ALL                  ALL                  NONE
```

Meaning of the output fields

STARTED
Specifies whether openFT is activated or not.

PROC-LIM
Maximum number of tasks that can be reserved simultaneously for the execution of FT requests.
Default setting following installation: 2

CONN-LIM
Maximum number of transport connections that can be reserved for asynchronous file transfer requests. Since each transport connection can only process one request at a time, CONN-LIMIT also defines the maximum number of requests that can be processed simultaneously. One third of the transport connections are reserved for requests from remote systems.
Default setting following installation: 16

ADM-CLIM
Maximum number of asynchronous administration requests including ADM traps that can be processed simultaneously.
Default setting following installation: 8

RQ-LIM
Maximum number of FT requests that can be entered at the same time in the request queue of the local system.
Default setting following installation: 2000

MAX-RQ-LIFE
Maximum number of days that an FT request is stored in the request file after its start time. When this period expires, the FT request is automatically removed from the request file.
Default setting following installation: 30

TU-SIZE
Maximum size of a transport unit in bytes. The load placed on the transport system by openFT can be controlled using this operand.
Default setting following installation: 65535

KEY-LEN
Current length of the RSA key. 0 means that encryption is deactivated. Default setting following installation: 2048

CCS-NAME
Name of the character set, which is used as standard character set for FT requests. The standard character set can be created with the CODED-CHARACTER-SET operand of the FTMODOPT command.
Default setting following installation: IBM1047

PTN-CHK
Defines whether or not enhanced sender checking is activated.
Default setting following installation: STD

DYN-PART
specifies whether dynamic partners are permitted (*ON) or not (*OFF).
Default setting following installation: ON

SEC-LEV
Local default value for the security level of the partner systems. This operand is only effective if FTAC functionality is being used. An important part of the access protection functions provided by this product lies in the allocation of security levels to remote systems. To this end, each system is allocated a security level designated using an integer in the range 1 to 100.

A default value for all remote systems is set by means of an operating parameter . All partners in the partner list for which the value STD is specified in the output of the FTSHWPTN command for SECLEV refer to this value.

This value is irrelevant for free dynamic partners (i.e. partner not entered in the partner list).
Default setting following installation: B-P-ATTR

FTAC-LOG
Scope for FTAC logging (ALL, MODIFY, REJECTED).
Default setting following installation: ALL

FT-LOG
Scope for FT logging (ALL, FAIL, NONE).
Default setting following installation: ALL

ADM-LOG
Scope of ADM logging (ALL, FAIL, MODIFY, NONE).
Default setting following installation: ALL

ENC-MAND
Specifies whether user data encryption is mandatory for openFT requests.
Default setting following installation: NO

OPENFT-APPL
Port number used by the local openFT. *STD means that the default port number 1100 is used.
The second line specifies whether the asynchronous inbound server is activated for openFT (ACTIVE), deactivated (DISABLED) or unavailable (INACT).
Default setting following installation: *STD

FTAM-APPL
Not relevant on z/OS systems; is always supplied with *NONE.
Default setting following installation: *NONE

FTP-PORT
Port number used by the local FTP server.
The second line specifies whether the asynchronous inbound server is activated for FTP (ACTIVE/DISABLED) or is unavailable or not installed (INACT/NAVAIL).
Default setting following installation: 21

ADM-PORT
Specifies the port number used by the local FT for remote administration. The default value is 11000.
The second line specifies whether the asynchronous inbound server is activated for remote administration requests (ACTIVE), deactivated (DISABLED) or unavailable (INACT).
Default setting following installation: 11000

HOST-NAME
Name of the host that is automatically taken over if you have specified a host during the FJGEN initialization run.
FTMODOPTDefault setting following installation: *NONE

IDENTIFICATION / LOCAL SYSTEM NAME
Instance identifier of the openFT instance currently set and the name of the local system. The instance identifieris used to identify the instance in the partner systems.
Default setting following installation: The value is formed from the value for FT-ID which is transferred with FJGEN: FJM<ftid> / $FJAM,FJM<ftid>

DEL-LOG
Specifies whether automatic deletion of log records is activated.
Default setting following installation: OFF
– ON: Day on which the records are to be deleted. A weekday (MON, TUE, WED, THU, FRI, SAT, SUN), a day of the month (1 through 31) or DAILY for daily deletion must be entered here.
  Default setting following installation: DAILY
– AT: Time (*hh:mm*) at which the records are to be deleted.
  Default setting following installation: 00:00
– RETPD: Minimum age of the records which are to be deleted (in days).
  Default setting following installation: 14

ADM-TRAP-SERVER
Name or address of the partner to which the ADM traps are sent.
*NONE means that the sending of ADM traps is deactivated.
Default setting following installation: *NONE

TRAP
This section with the rows CONS and ADM specifies the trap settings. The columns identify the events for which traps may be generated.
– SS-STATE: Subsystem state change (not for ADM traps)
– FT-STATE: State change of the openFT control process
– PART-STATE: Partner system state change
– PART-UNREA: Partner not reachable
– RQ-STATE: Request management state change
– TRANS-SUCC: Successfully completed requests
– TRANS-FAIL: Failed requests

  The possible values are ON or OFF.
  Default setting following installation: OFF (for all columns)

  The following rows specify the settings for the various trap types:

  CONS
  Settings for console traps FTR03XXX.

  ADM
  Setting for ADM traps to be output to the ADM trap server.

FUNCT
This section specifies the settings for monitoring (MONITOR) and tracing (TRACE).

The columns have the following meanings:
– SWITCH: Function activated (ON) or deactivated OFF
  Default setting following installation: OFF
– PARTNER-SELECTION: Selection according to protocol type of the partner system: ALL, OPENFT, FTP, ADM (only with TRACE), NONE
  Default setting following installation: ALL
– REQUEST-SELECTION: Selection according to request type: ALL, ONLY-ASYNC, ONLY-SYNC, ONLY-LOCAL, ONLY-REMOTE
  Default setting following installation: ALL
– OPTIONS (only with TRACE): NONE, NO-BULK-DATA (= minimal trace, i.e. no bulk data)
  Default setting following installation: NONE

The following rows specify what the settings apply to:

MONITOR
Setting for monitoring. Default setting following installation: OFF

TRACE
Setting for trace function. Default setting following installation: NONE

5.26 FTSHWPRF Display admission profile

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTSHWPRF, FTAC users can obtain information about their admission profiles. Either the contents of the selected admission profile or only its name can be output. It is not possible to use FTSHWPRF to access defined passwords or transfer admissions defined in the profile! If a transfer admission is forgotten, a new one must be specified using FTMODPRF.

Format

FTSHWPRF

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,SELECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
| **TRANS**FER**-ADM**ISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> /
| <c-string 8..32 with-low> / <x-string 15..64>
| **,OWN**ER**-ID**ENTIFICATION = ***OWN** / <name 1..8>

,INFORMATION = ***ONLY**-NAMES / ***ALL**

,OUTPUT = ***STDERR**(...) / ***STDOUT**(...)

***STDERR**(...) / ***STDOUT**(...)
| **LAYOUT** = ***STD** / ***CSV**

Operands

NAME =
Name of the admission profile you wish to view.

NAME = *ALL
Views all admission profiles.

NAME = <alphanum-name 1..8>
Views the admission profile with the specified name.

NAME = *STD
Displays the default admission profile for your own user ID.

SELECT-PARAMETER =
Selection criteria for the admission profiles you wish to view.

SELECT-PARAMETER = *OWN
Views all the admission profiles of which you are the owner. This means that you can view all the admission profiles which are assigned to your user ID.

SELECT-PARAMETER = *PARAMETERS(...)
Selection criteria with which you can access your admission profiles.

TRANSFER-ADMISSION =
Transfer admission defined in an admission profile as a selection criterion.

TRANSFER-ADMISSION = *ALL
TRANSFER-ADMISSION is not used as a selection criterion.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Only admission profiles for which no transfer admission has been specified are displayed.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Views the admission profile which can be addressed with this transfer admission.

OWNER-IDENTIFICATION =
Specifies, whose admission profiles you wish to view.

OWNER-IDENTIFICATION = *OWN
Views only your own admission profile.

OWNER-IDENTIFICATION = <name 1..8>
The FTAC user can only access his/her own admission profiles; the output corresponds to *OWN.

INFORMATION =
Scope of information desired.

INFORMATION = *ONLY-NAMES
FTAC only outputs the name of the admission profile and indicates whether it is privileged or blocked. An “∗“ is output for privileged profiles and a “!“ for blocked profiles.

INFORMATION = *ALL
FTAC outputs the contents of the admission profile, excluding any passwords and the transfer admission.

In the case of a blocked admission profile (marked with an "!" when output with INFORMATION=*ONLY-NAMES), the following values can appear in TRANS-ADM:

TRANS-ADM=	**Meaning**
(NOT-SPECIFIED)	No TRANSFER-ADMISSION specified in the admission profile.
(DUPLICATED)	The admission profile was blocked because the TRANSFER-ADMISSION was "detected" by another user and the profile was to be blocked in that case (USAGE=*PRIVATE is specified in the command FTCREPRF or FTMODPRF). "Detected" means that another user ID tried to assign the same TRANSFER-ADMISSION over again.
(LOCKED(by_user))	The admission profile was specifically blocked by the user (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(LOCKED(by_adm))	The admission profile was specifically blocked by the FTAC administrator (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(EXPIRED)	The validity of TRANSFER-ADMISSION has expired (EXPIRATION-DATE was specified in command FTCREPRF or FTMODPRF).

OUTPUT =
Output medium for the information.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon ";" (see section "Output in CSV format" on page 164).

Example

The user STEVEN wishes to view his admission profile UMSAWARE with the command FTSHWPRF to determine if the profile might endanger data protection:

```
FTSHWPRF␣NAME=UMSAWARE, -
      SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN),INFORMATION=*ALL
```

Short form:

```
FTSHWPRF␣UMSAWARE,(,STEVEN),*ALL
```

The output takes the following form:

```
UMSAWARE
 EXP-DATE    = 20121231
 IGN-MAX-LEV = (IBR)
 FILE        = UMSATZ
 USER-ADM    = (STEFAN,M4711,OWN)
 PROC-ADM    = SAME
 SUCC-PROC   = NONE
 FAIL-PROC   = NONE
 FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
               READ-FILE-DIRECTORY, FILE-PROCESSING)
 DATA-ENC    = YES
 LAST-MODIF  = 2012-07-11 13:38:11
```

The first line shows the name of the admission profile. EXP-DATE shows the expiration date of the admission profile. The next two lines show the settings which Steven made in the command FTCREPRF using the parameter IGNORE-MAX-LEVELS=(INBOUND-RECEIVE=*YES) and FILE-NAME= PROFIT. The values for USER-ADMISSION and PROCESSING-ADMISSION have not been set by Steven, but rather the default values have been used. The output SUCC-PROC=*NONE and FAIL-PROC=*NONE means that no follow-up processing is permitted. The output DATA-ENC=YES shows that Steven is especially careful, because this means that requests are only accepted if the user data is encrypted. Steven set this by using DATA-ENCRYPTION=*YES in the FTCREPRF command. The timestamp of the most recent change is shown under LAST-MODIF.

The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF only with the parameter NAME, but no other parameters.

Please note that as a rule not all properties of a profile are displayed. For example, optional parameters which do not differ from the default are not shown.

5.27 FTSHWPTN Display partner systems

Note on usage

User group: FT user and FT administrator

Functional description

The FTSHWPTN command is used to obtain the following information on partner systems included in the partner list of the current openFT instance:

- the names of the remote systems in the partner list,
- the status of the requests with the remote systems (activated or deactivated),
- priority assigned to the partner system,
- the setting for the openFT trace function on the partner system,
- the security level assigned to the remote system. Tthis security level applies only if FTAC functionality is used. The information can then also be obtained using the FTSHWRGE command.
- the number of not yet completed file transfer requests submitted in the local system,
- the number of file transfer requests submitted in the remote systems for the local system,
- the partner address.
- the type of sender checking,
- in the case of output in CSV format: also the time of the last access and the authentication level.

FTSHWPTN with the PARTNER=*ALL operand (default value) displays all **entered** dynamic partners. These can be recognized from the fact that they have no name. If you only want to output detailed information on one entered dynamic partner, you must specify the partner's address in the PARTNER operand.
In the case of the FTSHWPTN command openFT does not check whether an address is valid. If, for example, you specify a random address of a free dynamic partner, this will be displayed with the default properties of a free dynamic partner.

Format

FTSHWPTN

PARTNER = ***ALL** / <text 1..200 with-low>

,OUTPUT = ***STDERR**(...) / ***STDOUT**(...)

***STDERR**(...) / ***STDOUT**(...)

LAYOUT = ***STD** / ***CSV** / ***BS2-PROC** / ***ZOS-PROC**

,STATE = ***ALL** / ***ACTIVE** / ***DEACT** / ***INSTALLATION-ERROR** / ***NO-CONNECTION** / ***NOT-ACTIVE** / ***AUTOMATIC-DEACTIVATION** / ***INACTIVE-BY-AUTOMATIC-DEACT**

,INFORMATION = ***STD** / ***ALL**

Operands

PARTNER =
Partner system or systems about which information is to be output.

PARTNER = *ALL
Information on all partner systems is output.

PARTNER = <text 1..200 with-low>
Name or address of the partner system or group of partner systems about which information is to be output.
If you enter a name then you have two options:
You can either enter a unique partner name (1 - 8 alphanumeric characters) or a group of partners identified by a 1 to 7-character specification followed by an asterisk (*).
For more information on partner addresses, see section “Defining the partner computer” on page 99

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 164).

LAYOUT = *BS2-PROC
Output is supplied in the form of MODIFY-FT-PARTNER commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on BS2000.

LAYOUT = *ZOS-PROC
Output is supplied in the form of FTMODPTN commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on z/OS (.

STATE =
The scope of the output can be limited by the optional selection criteria in STATE. For an explanation of the selection criteria see page 301.

STATE = *ALL
The output is not limited by selection criteria.

STATE = *ACTIVE
All partner systems in the ACTIVE state are displayed.

STATE = *DEACT
All partner systems in the DEACT state are displayed.

STATE = *INSTALLATION-ERROR
All partner systems in the LUNK, RUNK, LAUTH, RAUTH, NOKEY and IDREJ state are displayed.

STATE = *NO-CONNECTION
All partner systems in the NOCON and DIERR state are displayed.

STATE = *NOT-ACTIVE
All partner systems not in the ACTIVE state are displayed.

STATE = *AUTOMATIC-DEACTIVATION
All partner systems are output which were assigned AUTOMATIC-DEACTIVATION.

STATE = *INACTIVE-BY-AUTOMATIC-DEACT
All partner systems are output which were actually deactivated using the option AUTOMATIC-DEACTIVATION.

INFORMATION = *STD / *ALL
Use this operand to control the scope of the information output. On *ALL, expanded address information is output, in addition to the standard information.

Example 1

Request information on all remote systems entered in the partner list:

Short output:

```
FTSHWPTN INF=*STD
NAME      STATE SECLEV    PRI  TRACE   LOC    REM P-CHK ADDRESS
          ACT    90       NORM FTOPT     0      0 FTOPT TEST011N
HOSTABS2  ACT   B-P-ATTR  NORM FTOPT     0      0 FTOPT HOSTABS2
HOSTBBS2  ACT   STD       NORM FTOPT     0      0 FTOPT HOSTBBS2
PCUSER    ACT    40       LOW  FTOPT     0      0 FTOPT %IP123.23.99.120
PC1       ACT    40       LOW  FTOPT     0      0 FTOPT PC1
UNIX1     ACT    50       HIGH FTOPT     0      0 FTOPT UNIX1
UNIX2     ACT    50       HIGH FTOPT     0      0 FTOPT UNIX2:102
FTPUX1    ACT   STD       NORM FTOPT     0      0       ftp://%IP132.19.122.50
```

Long output:

```
FTSHWPTN INF=*ALL
NAME      STATE SECLEV   PRI  TRACE  LOC  REM P-CHK ADDRESS
          INBND REQU-P                             ROUTING  IDENTIFICATION
          ACT   90       NORM FTOPT    0    0 FTOPT TEST011N
          ACT   STD                                       TEST011N
          ACT   STD
HOSTABS2  ACT   B-P-ATTR NORM FTOPT    0    0 FTOPT HOSTABS2
          ACT   STD                                       HOSTABS2.FUJI.NET
HOSTBBS2  ACT   STD      NORM FTOPT    0    0 FTOPT HOSTBBS2
          ACT   STD                                       HOSTBBS2.CLOUD.NET
          ACT   STD                                       ftamw.ftam2
          ACT   STD                                       ftamx.ftam3
PCUSER    ACT    40      LOW  FTOPT    0    0 FTOPT %IP123.23.99.120
          ACT   STD                                       %IP123.23.99.120
PC1       ACT    40      LOW  FTOPT    0    0 FTOPT PC1
          ACT   STD                                       PC1.FUSI.NET
UNIX1     ACT    50      HIGH FTOPT    0    0 FTOPT UNIX1
          ACT   STD                                       UNIX1.DREAM.NET
UNIX2     ACT    50      HIGH FTOPT    0    0 FTOPT UNIX2:102
          ACT   STD                                       %.UNIX2.$FJAM
FTPUX1    ACT   STD      NORM FTOPT    0    0       ftp://%IP132.19.122.50
          ACT   STD
```

The information displayed is explained below:

NAME
Symbolic names of the remote systems entered in the partner list.
This field remains empty for dynamic partners (see the first line in the example).

STATE
Status of the partner system.

ACT
The partner system is active.

DEACT
The partner system is deactivated.

NOCON
The transport connection setup failed.

LUNK
The local system is unknown on the remote FT system.

RUNK
The partner system is unknown on the local transport system.

ADEAC
The partner system is active. It is deactivated if the connection cannot be established. This state is only displayed if STATE=*AUTOMATIC-DEACTIVATION has been specified; otherwise, these partner systems are maintained under the ACT status.

AINAC
The partner system was deactivated following several unsuccessful attempts to establish a connection. This status is only possible if STATE=*AUTOMATIC-DEACTIVATION has been specified.

LAUTH
The local system could not be authenticated in the partner system. A current, public key of the local openFT instance must be made available to the partner system.

RAUTH
The partner system could not be authenticated in the local system. A current, public key of the partner system must be imported to the SYSKEY library.

DIERR
A data integrity error was detected on the connection to the partner system. This can be due either to an error in the transport system, or to manipulation attempts along the transfer route. The connection was terminated but the affected request was not (if it is restartable).

NOKEY
The partner does not accept a connection without encryption, but no key is present in the local system. A new key must be created using FTCREKEY.

IDREJ
The partner or a go-between instance does not accept the instance ID sent from the local system. You must check to see if the local instance ID is consistent with the entry in the partner's partner list.

SECLEV
Security level assigned to the remote system when it was entered in the partner list. These security levels apply only if the FTAC-BS2000 is also implemented. STD stands for the default security level set with the FTMODOPT command.

PRI
Priority of a partner with respect to the processing of requests. The possible values are NORM, LOW and HIGH.

TRACE
Trace setting. You may specify the values ON, OFF and FTOPT (if FTMODPTN is specified, TRACE=*BY-FT-OPTIONS).

LOC
Number of FT requests that have been submitted in the local system and that address the FT system specified with PARTNER.

REM
Number of FT requests that have been submitted in the remote FT system and addressed to the local FT system. The remote system is specified in PARTNER.

P-CHK
Type of sender checking for the current partner:

FTOPT
The global setting is valid.

T-A
The expanded sender checking is enabled for specific partners.

STD
The expanded sender checking is disabled for specific partners.

AUTH
With the aid of its public key in the SYSKEY library, the partner is subjected to an identity check (“authenticated”) by cryptographic means. The partner support the authentication level 2.

AUTH!
With the aid of its public key in the SYSKEY library, the partner is subjected to an identity check (“authenticated”) by cryptographic means. The partner support the authentication level 1.

NOKEY
No valid key is available from the partner system although authentication is required.

AUTHM
Authentication must be used.

ADDRESS
Partner address under which the remote system can be accessed. For more information on partner addresses, see section “Defining the partner computer” on page 99.

IDENTIFICATION
Instance ID of the partner (also see the FTADDPTN command in the System Administrator Guide).

ROUTING
SESSION-ROUTING-INFO of the partner, where required (also see the FTADDPTN command, in the System Administrator Guide).

INBND
State of the partner for inbound requests:

ACT
Inbound function is activated, i.e. requests issued remotely are processed.

DEACT
Inbound function is deactivated, i.e. requests issued remotely are rejected.

REQU-P
Operating mode for asynchronous outbound requests:

STD
Requests to this partner can be processed in parallel.

SERIAL
Requests to this partner are always processed serially.

5.28 FTSHWRGE Display partner systems

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The command FTSHWRGE is used to list the partner systems with which you can communicate by file transfer. In addition to indicating the name of the partner system, the security level is output which the FT administrator assigned to this system in the partner list. To determine which basic functions you are permitted to use, you must use the command FTSHWADS to obtain information on your admission set (see page 254).

Format

FTSHWRGE

USER-IDENTIFICATION = ***OWN** / <name 1..8>

,SELECT-PARAMETER = ***ALL** / ***PAR**AMETERS(...)

***PARA**METERS(...)
PARTNER = ***ALL** / <text 1..200 with-low>

,OUTPUT = ***STDERR**(...) / ***STDOUT**(...)

***STDERR**(...) / ***STDOUT**(...)
LAYOUT = ***STD** / ***CSV**

Operands

USER-IDENTIFICATION =
User ID for which you would like to have a list of accessible partner systems.

USER-IDENTIFICATION = *OWN
The FTAC user receives all the partner systems with which he/she can use at least one basic function.

USER-IDENTIFICATION = <name 1..8>
The FTAC user can only enter his/her own user ID here, the output corresponds to *OWN.

SELECT-PARAMETER =
Specifies selection criteria for the partner systems.

SELECT-PARAMETER = *ALL
Obtains information on all partner systems which can be reached.

SELECT-PARAMETER = *PARAMETERS(PARTNER = <text 1..200 with-low>)
Obtains information on this partner system. You can specify the name from the partner list or the address of the partner system. The following information is supplied:
– if you are permitted to communicate with this partner system.
– the security level assigned to this partner system.
For additional information to partner addresses, see section “Defining the partner computer” on page 99.

OUTPUT =
Output medium for the partner system listing.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output is in **C**haracter **S**eparated **V**alues format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by a semicolon “;“ (see section “Output in CSV format” on page 164).

Example

Steven Miller would like to find out about the security level of the computer BUYDACK. To do this, he uses the following command:

```
FTSHWRGE␣SELECT-PARAMETER=(PARTNER-NAME=BUYDACK)
```

Short form:

```
FTSHWRGE␣SEL=(BUYDACK)
```

He receives the following output:

```
SECLEV  PARTNER-NAME
  50    BUYDACK
```

The column SECLEV contains the security level of the partner system whose name appears in the PARTNER-NAME column.
If Steven had entered SELECT-PARAMETER=*ALL (or left out this parameter altogether), he would have received a similar but longer list of all accessible partner systems.

5.29 NCANCEL Cancel file transfer requests

Note on usage

User group: FT user and FT administrator

Alias name: FTCANREQ

Functional description

The NCANCEL command can be used to cancel a file transfer request or to abort the file transfer. The FT system deletes from the request queue the file transfer request that corresponds to the specified selection criteria and, if necessary, aborts the associated file transfer.

The following features apply to this command:

– FT requests submitted either in the local or the remote system can be canceled.

– A single command can be used to cancel several FT requests simultaneously.

– The FT requests to be canceled can be selected using different selection criteria.

– The FT user can only cancel file transfer requests, whose "owner" he/she is.

  The owner of an FT request submitted in the local system is the user ID under which the request was issued.
  The owner of an FT request submitted in the remote system is the user ID that is accessed in the local system for the request.

After the FT request is canceled, openFT initiates a follow-up processing in the event of failure (FAILURE-PROCESSING) which was previously specified in the NCOPY command. The following points apply:

– If you cancel a request issued in the local system, local FAILURE-PROCESSING will be initiated in any case; FAILURE-PROCESSING will be initiated in the remote system only if the data transfer process had already begun.

– If you cancel a request issued in a partner system, FAILURE-PROCESSING will be initiated both in the local and the remote system, respectively.

Note

– The file transfer requests aborted with NCANCEL remain in the request queue until both systems involved have informed each other of the abort action.

– Requests for which the file transfer proper has already been completed but where the decision to end the request has not yet been reached with the partner can no longer be canceled.

– If a request is canceled while pre-processing or post-processing is running in z/OS, openFT starts a separate "Cancel-Job" to terminate the processing job. This Cancel-Job is assigned a "Z" as the last letter in the job name in order to give it a higher priority than the processing jobs that are currently running.

Format

NCANCEL / **FTCANREQ**

TRANSFER-ID = ***ALL** / <integer 1..2147483647>

,**SEL**ECT = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

OWNER-IDENTIFICATION = ***OWN** / <name 1..8>

,**INIT**IATOR = (***LOC**AL, ***REMOTE**) / list-poss(2): ***LOC**AL / ***REMOTE**

,**PART**NER = ***ALL** / <text 1..200 with-low>

,**FILE-NAME** = ***ALL** / <filename 1..59> / <c-string 1..512 with-low>

You cannot issue an NCANCEL command without specifying any operands. In order to cancel or withdraw all your FT requests you may enter, for example:

```
NCANCEL *ALL
```

This is intended to prevent you unintentionally canceling all your FT requests by accidentally issuing an NCANCEL command without specifying any operands.

Operands

TRANSFER-ID =
Transfer ID of the FT request to be canceled.

TRANSFER-ID = *ALL
Deletes all FT requests if no further selection criteria are specified with SELECT. FT users can only delete FT requests of their own ID using this entry.

TRANSFER-ID = <integer 1..2147483647>
Request identification which was communicated to the local system in the FT request confirmation.

SELECT =
Contains selection criteria for FT requests to be canceled. A request is canceled if it satisfies all the specified criteria.

SELECT = *OWN
Cancels all FT requests associated with the own user ID and the specified TRANSFER-ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
Designates the owner of the FT requests. As an FT user you can omit this parameter, because you can only delete requests of your own ID.

OWNER-IDENTIFICATION = *OWN
Cancels only the FT requests under the user's own ID.

OWNER-IDENTIFICATION = <name 1..8>
As FT user you can only specify your own ID.

INITIATOR =
Initiator of the FT requests to be canceled.

INITIATOR = (*LOCAL,*REMOTE)
Cancels FT requests in the local system and in remote systems.

INITIATOR = *LOCAL
Cancels FT requests issued in the local system.

INITIATOR = *REMOTE
Cancels FT requests issued in remote systems.

PARTNER =
Cancels FT requests that were to be executed with a specific partner system.

PARTNER = *ALL
The name of the partner system is not used as a selection criterion to determine the FT requests to be canceled.

PARTNER = <text 1..200 with-low>
The FT requests that were to be executed with this partner are to be canceled.
The name must be specified in the same form in which it is output using NSTATUS.

FILE-NAME =
Cancels all FT requests in the local system that access this file or this library element whether as a send file or receive file. The file name or library member name must be specified exactly as it appears in the file transfer request.

FILE-NAME = *ALL
The file name is not used as a selection criterion to determine the FT requests to be canceled.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Cancels FT requests in the local system that access this file.

If multiple selection criteria are specified in the NCANCEL command, then each one of these must be valid for the requests that are to be canceled. Otherwise the NCANCEL command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria.
```

Example 1

An FT user wants to cancel all FT requests which carry his/her own ID. In order to do this, it is sufficient to issue the command only with the operand *ALL:

```
NCANCEL *ALL
```

If only one job was existent, openFT acknowledges the request with the following message:

```
FTR2072 Request 12334456 has been canceled
```

If several jobs were existent, all requests are deleted with no prompt for confirmation and cancellation is acknowledged by messages:

```
FTR2072 Request 12334558 has been canceled
FTR2072 Request 12334739 has been canceled
FTR2072 Request 12339336 has been canceled
```

Example 2

The FT request with the transfer ID 194578 is to be deleted. If the NCANCEL command is to be issued under the same ID as that under which the FT request was also submitted, the following command is sufficient:

```
NCANCEL TRANSFER-ID=194578
```

The recommended short form of this command is as follows:

```
NCANCEL 194578
```

Example 3

An FT user wishes to cancel all file transfer requests from remote system VAR001 that access his/her file DATA. This can be achieved with the following command:

```
NCANCEL TRANSDER-ID=*ALL,SELECT=(INITIATOR=*REMOTE,PARTNER=VAR001,
FILE-NAME=DATA)
```

The recommended short form of this command is as follows:

```
NCANCEL *ALL,SEL=(INIT=*REMOTE,PARTNER=VAR001,FILE=DATA)
```

5.30 NCOPY Transfer file asynchronously

Note on usage

User group: FT user

Alias name: FTACOPY

Functional description

The NCOPY command serves to transfer sequential files (PS data sets and generation data sets of this type), "entry sequenced" VSAM files, individual members of PO or PDSE data sets (libraries) and entire PO or PDSE data sets. In addition, openFT can also access migrated files in z/OS and transfer them to the remote system.

The local system is regarded as the system in which the command is issued, or in this case, the z/OS computer. The partner system is designated as the remote system.

5.30.1 Introduction to the NCOPY command

If you wish to transfer a file, you must first indicate whether you wish to send (TO) the file or receive (FROM) it by using the operand TRANSFER-DIRECTION.

Following this the PARTNER operand is used to define the system with which the transfer is to take place.

The next step is to define the characteristics of the local system by using the LOCAL-PARAMETER operand. The structure specifications for the LOCAL-PARAMETER are to be entered in parentheses, i.e. LOCAL-PARAMETER=(...).

The REMOTE-PARAMETER operand contains details of the remote system. The structure specifications for the REMOTE-PARAMETER must also be entered in parentheses, i.e. REMOTE-PARAMETER=(...). In addition, the partner system type may also be specified before these parentheses; the possible entries are *BS2000, *MSP (for a partner system with z/OS) or *ANY (see page 327).

The remaining “optional” operands (see page 338) are used to define the other characteristics of the file transfer, such as compressed or encrypted transfer or the starting time for the transfer.

5.30.1.1 The shortest form of the command

The mandatory parameters for the NCOPY command are the entries for
– direction of transfer
– name of the remote system
– name of the file in the local system (mandatory if FTAC-BS2000 is not implemented)
– name of the file in the remote system
– remote TRANSFER-ADMISSION.

A file transfer can be effected using these three parameters alone, if:
– the send and receive files are not password-protected

An example can be found on page 345.

This short command works because openFT assigns default values to all the values which are not specified. A detailed explanation of the abbreviations, order and default values of the operands can be found on page 316ff.

5.30.1.2 How to find out if the file transfer request has been executed

The command NSTATUS can be used to establish the status of file transfer requests that are not yet complete. On completion of a transfer, the result is stored in a logging record.

It is also possible to use the NCOPY command to request that a result message be generated. There are three ways of generating such a message:

– allow the result message to be created by the system,
– have a user-generated result message output as follow-up processing,

A system-generated message can only be requested in the local system. This is achieved using the LISTING operand which enables you for example to order a result list in all cases (LISTING=*PARAMETER(CONDITION=ANY)), or to order a result list only when the file transfer is aborted due to an error (LISTING=*PARAMETER(CONDITION=ON-FAILURE-ONLY)). The result list can be can be output to SYSLST or to a file. By default, no result list is created.

If a result list is printed then the user whose user ID was specified in the local TRANSFER-ADMISSION is informed of the termination of the file transfer job by means of an asynchronous message (NOTIFY message in z/OS).

Follow-up processing can also be requested in the NCOPY command. There are four types of follow-up processing:

– follow-up processing in the local system if the file transfer has been successfully completed
– follow-up processing in the remote system if the file transfer has been successfully completed

– follow-up processing in the local system if the file transfer has been aborted because of an error
– follow-up processing in the remote system if the file transfer has been aborted because of an error.

Follow-up processing after a successful file transfer can be defined for both systems by the operand SUCCESS-PROCESSING, while that following a failed file transfer is defined by FAILURE-PROCESSING. For details see the notes in the section section “Follow-up processing” on page 114.

If follow-up processing is to take place under a different user ID from that specified by TRANSFER-ADMISSION, then that user ID can be specified using the PROCESSING-ADMISSION operand.

Example

In this example described in the previous section the CLIST procedure 'CAESAR.MISTAKE.CLIST' is to be executed in the local system under the ID CAESAR with the account number ACCT0003 and the password #<gt>ABCDEFGH if the file transfer was not successful. In addition, a result list is to be printed under the ID FRED with account number ACCT0001 (without a password). If the file transfer has been successful, a message is to be sent to the user BERT in the remote system with account number ACCT0002 and password P1234567. In addition the file is to be assigned the name DATA and stored under the user ID BERT.

```
NCOPY
TRANSFER-DIRECTION=TO,
PARTNER=VAR001,
LOCAL-PARAMETER=(FILE=DATA,
TRANSFER-ADMISSION=(USER-
IDENTIFICATION=ANTON,ACCOUNT=ABRE0001,PASSWORD=HUGO),
PROCESSING-ADMISSION=(USER-IDENTIFICATION=CAESAR,
ACCOUNT=ABRE0003,PASSWORD=ABCDEFGH),
FAILURE-PROCESSING='EX IRRTUM',
LISTING=*SYSLST),
REMOTE-PARAMETER=*MSP(FILE=DATEN,
TRANSFER-ADMISSION=(USER-IDENTIFICATION=BERT,
ACCOUNT=ABRE0002,PASSWORD=P1234567),
SUCCESS-PROCESSING='SEND ''FILE TRANSFER O.K.'',USER(*)')
```

A possible short form of this command is as follows:

```
NCOPY TRANS=TO,PARTNER=VAR001,
LOC=(FILE=DATA,TRANS=(ANTON,ABRE0001.HUGO),
PROC=(CAESAR,ABRE0003,ABCDEFGH),FAIL='EX IRRTUM'),
REM=*MSP(FILE=DATEN,TRANS=(BERT,ABRE0002,P1234567),
SUCC='SEND ''FILE TRANSFER O.K.'',USER(*)')
```

The commands defining follow-up processing must be enclosed in single quotes. Any quotes specified within the command must be doubled (see SEND command in the example above).

If neither a result list nor follow-up processing has been requested for a request, you can use the logging function to determine whether the request has been carried out. A request may generate up to four logging records:

– an FT logging record in the initiator system on request termination. A precondition for this is that the request has been correctly accepted by openFT and FT logging is active.

– an FT logging record in the responder system on request termination. A precondition for this is that the actual file transfer operation has already been concluded and FT logging is active.

– an FTAC logging record in the initiator system on acceptance of the request. A precondition for this is that openFT-AC is used in the initiator.

– an FTAC logging record in the responder system on acceptance of the request. A precondition for this is that openFT-AC is used in the responder.

You view logging records with FTSHWLOG (see page 259).

5.30.2 Full form of the NCOPY command

Format

(part 1 of 4)

NCOPY / **FTACOPY**

TRANSFER-**DIR**ECTION = **TO-**PARTNER / **FROM-**PARTNER

,**PART**NER = <text 1..200 with-low>

,**LOC**AL-PARAMETER = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **FILE-NAME** = ***NOT-SPECIFIED** / <filename 1..59> / <c-string 1..512 with-low>
| ,**PASS**WORD = ***NONE** / <alphanum-name 1..8>
| ,**TRANS**FER-**ADM**ISSION = ***SAME** / <alphanum-name 8..32> / <x-string 15..64> /
| <c-string 8..32 with-low> / ***PAR**AMETERS(...)
| ***PAR**AMETERS(...)
| | **USER-ID**ENTIFICATION = <name 1..8>
| | ,**ACCOUNT** = ***NONE** / <alphanum-name 1..40> / c-string 1..40>
| | ,**PASS**WORD = ***NONE** / <alphanum-name 1..8>
| ,**PROC**ESSING-**ADM**ISSION = ***SAME** / ***NOT-SPECIFIED** / ***PAR**AMETERS(...)
| ***PAR**AMETERS(...)
| | **USER-ID**ENTIFICATION = <name 1..8>
| | ,**ACCOUNT** = ***NONE** / <alphanum-name 1..40> / c-string 1..40>
| | ,**PASS**WORD = ***NONE** / <alphanum-name 1..8>
| ,**SUCC**ESS-PROCESSING = ***NONE** /<c-string 1..1000 with-low>
| ,**FAIL**URE-PROCESSING = ***NONE** / <c-string 1..1000 with-low>
| ,**LIST**ING = ***NONE** / ***SYSLST** / ***LISTFILE** / ***PAR**AMETERS(...)
| ***PAR**AMETERS(...)
| | **OUT**PUT = ***SYSLST** / ***LISTFILE**
| | **,CONDITION** = ***ANY** / ***ON-FAILURE-ONLY**
| ,**CODED-CHAR**ACTER-**SET** = ***STD** / <alphanum-name 1..8>

(part 2 of 4)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)
    *BS2000(...)
        FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <c-string 1..512 with-low> /
                    *LIBRARY-ELEMENT(...)
          *LIBRARY-ELEMENT(...)
              LIBRARY = *NOT-SPECIFIED / <filename 1..54>
              ,ELEMENT = *NOT-SPECIFIED /
                         <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
                  <filename>(...) / <composed-name>(...)
                      VERSION = *STD / <text 1..24>
              ,TYPE = *NOT-SPECIFIED / <name 1..8>
        ,PASSWORD = *SAME / *NONE / <c-string 1..4> / <x-string 1..8> /
                    <integer -2147483648..2147483647>
        ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /
                              <c-string 8..32 with-low> / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <name 1..8>
              ,ACCOUNT = *NONE / <alphanum-name 1..8>
              ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
        ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <alphanum-name 1..8>
              ,ACCOUNT = *NONE / <alphanum-name 1..8>
              ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
        ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
        ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
        ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 3 of 4)

```
*MSP(...)
|  FILE-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low>
|  ,PASSWORD = *NONE / <alphanum-name 1..8>
|  ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /  <c-string 8..32 with-low> /
|                                *PARAMETERS(...)
|     *PARAMETERS(...)
|     |  USER-IDENTIFICATION = <name 1..8>
|     |  ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
|     |  ,PASSWORD = *NONE / <alphanum-name 1..8>
|  ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
|     *PARAMETERS(...)
|     |  USER-IDENTIFICATION = <name 1..8>
|     |  ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
|     |  ,PASSWORD = *NONE / <alphanum-name 1..8>
|  ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
|  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
|  ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
*ANY(...)
|  FILE-NAME = *NOT-SPECIFIED / <c-string 1..512 with-low> /
|                 *LIBRARY-ELEMENT(...)
|     *LIBRARY-ELEMENT(...)
|     |  LIBRARY = *NOT-SPECIFIED / <c-string 1..63 with-low>
|     |  ,ELEMENT = *NOT-SPECIFIED / <c-string 1..64 with-low>(...)
|     |     <c-string 1..64 with-low>(...)
|     |     |  VERSION = *NONE / *STD / <c-string 1..24 with-low>
|     |  ,TYPE = *NONE / *NOT-SPECIFIED / <c-string 1..8 with-low>
|  ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
|  ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
|                                <c-string 8..32 with-low> / *PARAMETERS(...)
|     *PARAMETERS(...)
|     |  USER-IDENTIFICATION = <c-string 1..67 with-low>
|     |  ,ACCOUNT = *NONE / <c-string 1..64 with-low>
|     |  ,PASSWORD = *NONE / <c-string 1..64 with-low> /  <x-string 1..128 with-low>
|  ,PROCESSING-ADMISSION = *SAME / *NONE / *PARAMETERS(...)
|     *PARAMETERS(...)
|     |  USER-IDENTIFICATION = <c-string 1..67 with-low>
|     |  ,ACCOUNT = *NONE / <c-string 1..64 with-low>
|     |  ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
|  ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
|  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
|  ,CODED-CHARACTER-SET = *STD / <c-string 1..8 with-low>
```

(part 4 of 4)

```
,COMPRESS = *NONE / *BYTE-REPETITION / *ZIP
,WRITE-MODE = *REPLACE-FILE / *NEW-FILE / *EXTEND-FILE
,DATA-TYPE = *NOT-SPECIFIED / *CHARACTER (...) / *BINARY (...) / *USER
    *CHARACTER(...)
        TRANSPARENT = *NO / *YES
    *BINARY(...)
        TRANSPARENT = *NO / *YES
,PRIORITY = *NORMAL / *HIGH / *LOW
,START = *SOON / *EARLIEST(...)
    *EARLIEST(...)
        DATE = *TODAY / *TOMORROW / <date 8..10>
        ,TIME = 00:00 / <time 1..8>
,CANCEL = *NO / *AT(...)
    *AT(...)
        DATE = *TODAY / *TOMORROW / <date 8..10>
        ,TIME = 23:59 / <time 1..8>
,DATA-ENCRYPTION = *NO / *YES / *ONLY-DATA-INTEGRITY
,RECORD-SIZE = *NOT-SPECIFIED /<integer 1..32756>
,RECORD-FORMAT = *STD / *FIXED / *VARIABLE / *UNDEFINED
,TABULATOR = *AUTO / *ON / *OFF
,TARGET-FILE-FORMAT = *SAME / *BLOCK-ORIENTED / *SEQUENTIAL(...)
    *SEQUENTIAL(...)
        RECORD-FORMAT = *SAME / *UNDEFINED
```

Operands

TRANSFER-DIRECTION =
Direction of transfer.

TRANSFER-DIRECTION = TO-PARTNER
The local system is the send system. The files are dispatched to the partner system.

TRANSFER-DIRECTION = FROM-PARTNER
The local system is the receive system. The files are obtained from the partner system.

PARTNER = <text 1..200 with-low>
Name of the partner system as defined by the FT administrator in the partner list or the address of the partner system. For more information on address specifications, see section “Defining the partner computer” on page 99.

Specifications for the local system (LOCAL-PARAMETER)

LOCAL-PARAMETER = *PARAMETERS(...)
Specifications for the local system.

FILE-NAME =
Name of the file entry in the library in the local system (send file or receive file).

FILE-NAME = *NOT-SPECIFIED
The name of the file is known locally because it has already been completely defined in the FTAC admission profile addressed locally.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
When sending, the name of the file or pre-processing command, or, when receiving, the name of the post-processing command. The specifications differ for with and without pre- and post-processing.

*Specifications **without** pre- or post-processing on FILE-NAME*

All types of filename can be either fully qualified or partially qualified:

– Fully qualified specification: the filename is enclosed in single quotes; the "first level qualifier" is the user ID or the alias under which the file is or will be cataloged.

– Partially qualified specification: the filename is not enclosed in single quotes; in this case, openFT adds the user ID for which the file transfer is being performed (TRANSFER-ADMISSION operand) as the "first level qualifier".

– If you use the c-string data type for a partially qualified filename, you must specify it in the form C'FILE.XYZ'.

If complete PO or PDSE data sets are to be transferred, the receive file only needs to be specified in the request with the corresponding type :O: or :E: openFT with a version < 10 is running on the partner system.

openFT permits the automatic generation of unique filenames as a simple way of preventing conflicts. You do this by entering %UNIQUE in the filename (see section "Unique file names for receive files" on page 52).

*Specifications **with** pre- or post-processing on FILE-NAME*

– If you specify pre-processing commands on a send operation, the specified commands are first started as a TSO job. The data is output via the %TEMPFILE variable or via SYSPRINT to a temporary file ("pre-processing")

– If you specify post-processing commands on a receive operation, openFT provides the first command with the transferred data via the %TEMPFILE variable or via SYSUT1 and waits until processing is concluded ("post-processing").

For both pre- and post-processing, a c-string must be specified on FILE-NAME. The first character must be a pipe symbol ‘|’, followed by the command string. If several commands are specified, they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME=C'|Command1;Command2;Command3; ...'
```

You should construct command sequences using the TSO WHEN command, e.g.:

```
command1;WHEN SYSRC(< 12) command2;WHEN SYSRC(< 12) command3;...
```

The total maximum length of commands is restricted to the maximum file name length. For more information refer to the section “File transfer with preprocessing, postprocessing and follow-up processing” on page 33. Also refer to the topic “Pre-processing” at the example starting on page 345.

If an error occurs during command execution, transfer is aborted with message FTR2206 or FTR2207.

If a transfer request with pre- and post-processing is to be restartable, the characters ‘|&’ must be specified instead of ‘|’. For more details, also see section “File transfer with preprocessing, postprocessing and follow-up processing” on page 33.

Example

```
FILE-NAME = C'|&command1;command2;command3; ...'
```

To prove admission for pre-processing or postprocessing, the local TRANSFER-ADMISSION must either be explicitly supplied with USER-ID, ACCOUNT and PASSWORD or implicitly supplied with these specifications via a transfer admission belonging to an admission profile that contains them.

PASSWORD =
Password authorizing access to the file in the local system. If the file in the local system is password-protected (by means of the TSO command "PROTECT") , the password must be specified in this operand as:

– a write password for a receive file, or
– a read password for a send or receive file that is not protected by a write password but by a read password,

Newly created receive files are not given a password by this operand. PASSWORD is ignored in such cases.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password authorizing access to the file in the local system.

TRANSFER-ADMISSION =
Transfer admission of the user for the local system.

TRANSFER-ADMISSION = *SAME
The ID of the user entering the command is valid for the file transfer.

To prove admission for pre-processing or post-processing, the local TRANSFER-ADMISSION must either be explicitly supplied with USER-ID, ACCOUNT and PASSWORD or implicitly supplied with these specifications via a transfer admission belonging to an admission profile that contains them. The same applies if migrated files are to be transferred to a remote system. If the admission is missing in the case of migrated files, message FTR2029 is issued.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Only if FTAC functionality is used can the file name for the local system be defined in an FT profile. The transfer admission defined in the FT profile must be specified here. From this transfer admission the access rights in the local system can be defined. These access rights are also defined in the FT profile. The alphanumeric entry is converted internally to lowercase characters.

The alphanumeric entries are converted internally to lowercase characters.

When entering the transfer admission in the form of a c-string you must remember that in z/OS a transfer admission defined as a c-string is stored using uppercase characters in the profile. To go with this you must also enter it in the NCOPY command, to ensure that conversion to lowercase is avoided. Here, you should bear in mind the notes regarding the distinction between uppercase and lowercase characters which can be found on page 153.

TRANSFER-ADMISSION = *PARAMETERS(...)
User ID, account number and password under which file transfer in the local system is to be performed. The operands in parentheses can also be used a positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8>
User ID in the local system.

ACCOUNT = _
Account number under which file transfer is performed in the local system.

ACCOUNT = *NONE
The default account number of the user ID is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Account number of the user in the local system.

PASSWORD =
Password authorizing the user to access the local system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password that authorizes the user to access the local system.

PROCESSING-ADMISSION =
Contains information concerning the authorization of a user in the local system to perform follow-up processing.

PROCESSING-ADMISSION = *SAME
The relevant TRANSFER-ADMISSION values (see above) of the local system apply. This specification is only of value if the local TRANSFER-ADMISSION has either been explicitly supplied with USER-ID, ACCOUNT and PASSWORD or has been supplied implicitly via the transfer admission of an admission profile that contains these specifications. Otherwise the PROCESSING-ADMISSION must itself explicitly contain these specifications, either through their entry in this NCOPY command or through the use of a corresponding admission profile.

PROCESSING-ADMISSION = *NOT-SPECIFIED
Only if FTAC functionality is used can the entry for PROCESSING-ADMISSION be predefined by an FT profile. This entry must not be specified in the FT request.

PROCESSING-ADMISSION = *PARAMETERS(...)
User ID, account number and password of the user for whom the follow-up processing is to be performed. The operands in parentheses can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8>
User ID in the local system. This ID must be specified in the syntax of the local z/OS system.

ACCOUNT = *NONE
Access is possible without a password.
The default account number of the user ID specified in the USER-IDENTIFICATION is used, see page 126.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Account number or "accounting information" of the user in the local system. The account number must be specified in the syntax of the local system.

PASSWORD =
Password authorizing the user to access the local system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password that authorizes the user to access the local system.

Follow-up processing in the local system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING. The individual commands must be separated by a semicolon. If a character string is enclosed in single or double quotes (’ or ") within a command sequence, openFT will not interpret any semi-colon present within this character string as a separator.

Example

```
SUCC = ’command1;command2;command3’
```

The entries for the operands SUCCESS-PROCESSING and FAILURE-PROCESSING may total up to 1000 characters. If the length limit for follow-up processing data is exceeded due to variable replacement, openFT ends the request and returns an error message.

The commands/JCL statements must comply with the z/OS syntax rules; openFT does not check the syntax of the commands and statements specified for follow-up processing.

The character string enclosed in single quotes may consist of:

– one or more TSO commands separated from one another by semicolons: 'command' or 'command;command;command...'

– one or more JCL statements separated from one another by semicolons: 'statement' or 'statement;statement;statement...'
  You can specify the following JCL statements which are identified by their JCL identifier:
  // "real" JCL statements
  //* comment lines
  no JCL identifier: in-stream data

– the specific openFT statement 'ALLOC DSNAME(filename)', if a user-created PS data set containing all JCL statements for follow-up processing is to be allocated or

– the specific openFT statement 'ALLOC DSNAME (filename(member-name))' if a user-created PO or PDSE data set is to be allocated whose member "member-name" contains all JCL statements for follow-up processing.

The specific openFT statement 'ALLOC DSNAME(...' must be entered exactly as specified above.

If the TSO command ALLOC is to be specified as follow-up processing, then - unlike the special openFT statement ALLOC DSNAME - the full TSO command name must be specified (i.e. ALLOCATE DSNAME ...).

If one or more TSO commands or one or more JCL statements are specified as follow-up processing, then openFT generates a job that is passed to the system via the internal reader. The FT administrator can adapt uniform job envelopes that are based on the follow-up processing commands surrounding it to the circumstances of the local system.

Use of variables in follow-up processing

Variables can be used both in the TSO commands and JCL statements specified for follow-up processing, in the job envelope for follow-up processing and in the statements of the jobs which are to be started with the openFT statement ALLOC DSNAME as follow-up processing. Before the start of follow-up processing, openFT replaces the variables with the corresponding values from the specifications in the command, or %RESULT with the message code of the request. Then the commands of the follow-up processing are executed.

For a description of the variables that are permitted in z/OS, see page 116.

If one of the above-named symbolic identifiers remains in this form, that is without replacement, then the initial percentage sign must be doubled, as in %%FILENAME.

A programmer's name can be specified with the other specifications for follow-up processing; for more information, refer to page 122.

SUCCESS-PROCESSING =
Follow-up processing to be executed in the local system after a successful file transfer.

SUCCESS-PROCESSING = *NONE
No follow-up processing to be executed.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
z/OS commands to be executed in the local system after successful file transfer.

FAILURE-PROCESSING =
Follow-up processing to be carried out in the local system if an error is detected after setting up the link or during the file transfer.

FAILURE-PROCESSING = *NONE
No follow-up processing to be executed.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
z/OS commands to be executed in the local system after unsuccessful file transfer. The same specifications are hold for FAILURE-PROCESSING as for SUCCESS-PROCESSING, see above.

LISTING =
Requests a result list in the local system. This listing is generated for the user for whom file transfer is performed. The default setting for LISTING in the local parameters is *NONE.
If the result list is printed, the job initiated by openFT for this purpose causes an asynchronous end message to be sent to the user whose user ID is specified in the local TRANSFER-ADMISSION (JOB parameter NOTIFY).

LISTING = *NONE
No result list is generated.

LISTING = *SYSLST
A result list is printed out.

LISTING = *LISTFILE
openFT creates a result list and stores it under the ID specified in the local TRANSFER-ADMISSION operand. This file has the following name:

inst.T*transfer-id*.LST (the user ID is the "first level qualifier")

Here, *inst* stands for the instance name of the current openFT instance and t*ransfer-id*. for the identification number of the FT request.

If the "transfer-id." is longer than 7 characters, it is divided into two parts in this filename as follows: instance name.Tpart1-id.Tpart2-id.LST, where "part1-id" always has a length of 7 characters and "part2-id" always has a length of 1 to 3 characters. The file is created on the same volume as non-existent receive files. The FT administrator can specify the volume for these (the same for all transfer requests). If the FT administrator has not specified a volume, the system defaults for newly created files apply (see also the section “Result lists generated by openFT for z/OS” on page 110).

LISTING = *PARAMETERS(...)
Requests a result list in the local system. The list is created for the user for whom file transfer is performed.

OUTPUT =
Output medium.

OUTPUT = *SYSLST
The result list is printed out.

OUTPUT = *LISTFILE
openFT stores the result list under the ID specified in the local TRANSFER-ADMISSION operand. This file has the following name:

inst.Ttransfer-id.LST (the user ID is the "first level qualifier")

For more information see LISTING=*LISTFILE on page 326.

CONDITION =
Condition under which a result list should be generated.

CONDITION = *ANY
A result list is generated in every case.

CONDITION = *ON-FAILURE-ONLY
A result list is only generated when the file transfer is aborted with an error.

CODED-CHARACTER-SET =
Coding (character set) that is to be used to read or write the local file.

CODED-CHARACTER-SET = *STD
The character set used by default to read or write the local file is the character set specified globally with FTMODOPT or defined in the FT parameter library.

CODED-CHARACTER-SET = <alphanum-name 1..8>
Coding that is to be used to read or write the local file. The character set must be known in the local system.

Specifications for the remote system (REMOTE-PARAMETER)

REMOTE-PARAMETER =
Contains information about or for the remote system. This entry specifies the type of remote system. It also defines the syntax in which the remote system expects the value assignments.

REMOTE-PARAMETER = *BS2000(...)
The value assignments for the remote system are given in BS2000 syntax. The local system checks whether the specified values conform to this syntax.

REMOTE-PARAMETER = *MSP(...)
The value assignments for the remote system are in the syntax of the OS/390 or MVS system. The local system checks if the values specified conform to this syntax.

REMOTE-PARAMETER = *ANY(...)
The local system does not check the syntax in which the value assignments for the remote system are specified. Value assignments for the local system cannot be used as default values for the remote system. The value assignments must be in quotation marks. Double quotes must be used for any quotation marks within single quotes (e.g. PASSWORD= ’C”ABCD”’).

FILE-NAME =
Name of the file or the library in the remote system (send file or receive file). It must be specified in the syntax and conform to the conventions of the remote system.

REMOTE-PARAMETER=	*BS2000	*MSP	*ANY
relevant for:	X	X	X
default value:	*NOT-SPECIFIED	*NOT-SPECIFIED	*NOT-SPECIFIED

openFT partners as of V7.0 offer an option with which unique file names can be generated automatically in order to easily prevent conflict situations. This is achieved specifying the string %UNIQUE in the file name (see section “Unique file names for receive files” on page 52 for details).

FILE-NAME = *NOT-SPECIFIED
relevant for *BS2000, *MSP and *ANY:
Only if FTAC functionality is used in the remote system can the file name be predefined, either partially or completely, in an FT profile. The file name or partial file name does not then have to be known to the request submitter. The file name may not be specified in the command.

FILE-NAME = <filename 1..54> / <filename 1..59> / <c-string 1..512 with-low>
Name of the file or pre-processing command, when receiving, or of the post-processing command, when sending. The specifications differ for **with** and **without** pre- and post-processing.

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
FILE-NAME=	<filename 1..54> <c-string 1..512 with-low> *POSIX(NAME = <posix-pathname 1..510>	<filename 1..59> <c-string 1..512 with-low>	<c-string 1..512 with-low>

*Specifications **without** pre- or post-processing on FILE-NAME*

Name of the file in the remote system (send or receive file):

– With *BS2000 and *MSP, this file name can be specified without a user ID if the file is cataloged under the user ID for which the file transfer is performed (TRANSFER-ADMISSION operand).
– This file name must be specified with a user ID ($userid.filename) if the file is not cataloged under the user ID for which the file transfer is performed (TRANSFER-ADMISSION operand).
– With FILE =<filename 1..59>, it is also possible to address library elements in z/OS provided that the FT product used in the partner system supports the transfer of library elements. For a BS2000 partner, you should use the structure *LIBRARY-ELEMENT.
– If you use the c-string data type for a partially qualified filename in a z/OS partner system (*MSP), you must specify it in the form C'FILE.XYZ'.
– If complete PO or PDSE data sets are to be transferred, the receive file must always be specified with the corresponding type :O: or :E: in the request, provided that the partner is still openFT on z/OS with a version < V10.
– If an FT product is used in the remote BS2000 and this product carries out the customary BS2000 extension of file names of the form *$filename* to include the standard user ID, the file name may be specified in this form. If this is not the case an error will result.

*Specifications **with** pre- and post-processing on FILE-NAME*

If you specify a pre-processing command when receiving, the result from the pre-processing command is sent to the remote system's standard output (BS2000/OSD: SYSLST; z/OS: SYSPRINT) before being transferred. You can also address the output from the pre-processing command via the %TEMPFILE variable. The advantage of this is that the output can have any file format and the file is transferred in this format. If you do not specify %TEMPFILE then the output must have a format which is permitted at the remote system's standard output, i.e. in BS2000/OSD systems it must take the form of a SAM-V file. On z/OS, this is a PS file with a variable block size.

If you specify a post-processing command when sending, the transferred file is used as input for the post-processing command. This file can be addressed with the variable %TEMPFILE. If %TEMPFILE is not specified, read-in is done via the standard input (BS2000: SYSDTA, z/OS:SYSUT1). If the remote system is a BS2000/OSD, the file must be a SAM-V or ISAM-V file. On z/OS, this is a PS file with a variable block size.

For both pre- and post-processing, a c-string must be specified on FILE-NAME. The first character must be a pipe symbol '|', followed by the command string. If several commands are specified, they must be separated by a semicolon (';').

Example

```
FILE-NAME=C'|Command1;Command2;Command3; ...'
```

The maximum length of the entire command is limited to the maximum length of the file name. You will find more detailed information on this in the section "File transfer with preprocessing, postprocessing and follow-up processing" on page 33.

If an error occurs while executing the commands , the transfer is aborted and the message FTR2206 or FTR2207 appears.

If a transfer request with pre- and post-processing is to be restartable, the characters '|&' must be specified instead of '|'. For more detailed information, also see section "File transfer with preprocessing, postprocessing and follow-up processing" on page 33.

Example

```
FILE-NAME = C'|&Command1;Command2;Command3; ...'
```

FILE-NAME = *LIBRARY-ELEMENT(...)

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
only relevant for:	X	[1]	X
default value:			*NOT-SPECIFIED

[1] For z/OS systems, library members must be defined with FILE-NAME=.

Specifies that a library member is to be transferred. *NOT-SPECIFIED for all three operands is invalid, because the entry would not guarantee access to a library member in the remote system.

Furthermore, the remote system must be capable of processing library members.

LIBRARY =
Name of the library in the remote system.

LIBRARY = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the name of the library be predefined in an FT profile. The name of the library must not then be made known to the request submitter, nor may it be specified in the command.

LIBRARY = <filename 1..64> / <c-string 1..512 with-low>
relevant for *ANY.
Name of the library in the remote system. It must be specified in the conventions of the remote system.

LIBRARY = <filename 1..54> / <c-string 1..512 with-low>
relevant for *BS2000.
Name of the library in the remote system. It must be specified in the conventions of the remote system.

ELEMENT =
Name of the library member in the remote system.

ELEMENT = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the name of the library member be predefined in an FT profile. The name of the library member must not then be made known to the request submitter, nor may it be specified in the command.

ELEMENT = <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>
relevant for *BS2000.

Name of the library member in the remote system. It must be specified in the conventions of the remote system.

ELEMENT = <c-string 1..64 with-low>(...)
relevant for *ANY.
Name of the library member in the remote system. It must be specified in the conventions of the remote system.

VERSION =
Version of the member in the remote system.

VERSION = *NONE
relevant for *ANY.
No specification of the version in the remote system must be made.

VERSION = *STD
relevant for *BS2000 and *ANY.
Highest version of the member

VERSION = <text 1..24>
relevant for *BS2000.
Version of the member.

VERSION = <c-string 1..24 with-low>
relevant for *ANY.
Version of the member. It must conform to the conventions of the remote system.

TYPE =
Member type in the remote system.

TYPE = *NONE
relevant for *ANY.
The member type does not have to be specified in the remote system.

TYPE = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the library member type be predefined in an FT profile. The type of the library member must not then be made known to the request submitter, nor may it be specified in the command.

TYPE = <name 1..8>
relevant for *BS2000.
Member type in the remote system. It must be specified in the conventions of the remote system.

TYPE = <c-string 1..8 with-low>
relevant for *ANY.
Member type in the remote system. It must be specified in the conventions of the remote system.

PASSWORD =

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:		*NONE	*NONE

Password authorizing access to the file in the remote system. The file password must be specified in the remote system's syntax and conform to the conventions of the remote system.

If the file in the remote system is protected with a password, the password must be specified in this operand as:

– a write password for a receive file, or
– a read password for a send or receive file that is not protected by a write password but by a read password, or
– a password for the execution of a send or receive file that is protected neither by a read nor by a write password but by an execute command.

Newly-created receive files are not given a password by this operand. PASSWORD is ignored in such cases.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
relevant for *BS2000.
BS2000 file password.

PASSWORD = <alphanum-name 1..8>
relevant for *MSP.
OS/390 or z/OS file password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128>
relevant for *ANY.
With *ANY, the file password must always be in inverted commas.

TRANSFER-ADMISSION =
Contains information on authorization to perform file transfers in the remote system.

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
relevant for:	X	X	X
default value:	mandatory	mandatory	*NONE

TRANSFER-ADMISSION = *NONE
relevant for *ANY.
The remote system does not require/recognize any transfer admission.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low>
relevant for *BS2000, *MSP and *ANY.
When FTAC functionality is used in the remote system, only the TRANSFER-ADMISSION predefined in the admission profile may be specified. The alphanumeric entries are converted internally to lowercase letters.
When entering the transfer admission in the form of a c-string you should bear in mind the notes regarding the distinction between uppercase and lowercase characters which can be found on page 153. With an z/OS partner system, you must remember that a transfer admission defined as a c-string is stored in the profile using uppercase characters. To go with this you must also enter it in the NCOPY command, to ensure that conversion to lowercase is avoided.

TRANSFER-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system for which the follow-up processing is to be performed. The operands in parentheses can be used as positional operands without their keywords.

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
USER-IDENTIFICATION =	<alphanum-name 1..8>	<name 1..8>	<c-string 1..67 with-low>
ACCOUNT=	*NONE <alphanum-name 1..8>	*NONE <alphanum-name1..40> / <c-string 1..40>	*NONE <c-string 1..64 with-low>
PASSWORD=	*NONE <c-string 1..8> / <c-string 9..32> <x-string 1..16>	*NONE <alphanum-name 1..8>	*NONE <c-string 1..64 with-low> / <x-string 1..128>

USER-IDENTIFICATION =
relevant for *BS2000, *MSP and *ANY.
Identification of the user (user ID) in the remote system.

ACCOUNT =
relevant for *BS2000, *MSP and *ANY.
Account number of the user in the remote system.

ACCOUNT = *NONE
relevant for **BS2000, *MSP and ANY.
The remote system does not require an account number.

PASSWORD =
relevant for *BS2000, *MSP and *ANY.
Password authorizing the user to access the remote system.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

PROCESSING-ADMISSION =
relevant for *BS2000, *MSP and *ANY.
Contains information about a user's authorization to perform follow-up processing in the remote system.

FTP partners do not support remote follow-up processing.

PROCESSING-ADMISSION = *SAME
The relevant REMOTE TRANSFER-ADMISSION values apply.

PROCESSING-ADMISSION = *NONE
relevant for *ANY.
.No transfer admission is required for follow-up processing. See also the description of PROCESSING-ADMISSION=*NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-SPECIFIED
Only if FTAC functionality is used in the remote system can the PROCESSING-ADMISSION be predefined in an FT profile. It must not then be made known to the request submitter, nor may it be specified in the command.

PROCESSING-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system, for which the follow-up processing is to be performed. The parameters in parentheses can be used as positional operands without their keywords.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
USER-IDENTIFICATION =	<alphanum-name 1..8>	<name 1..8>	<c-string 1..67 with-low>
ACCOUNT =	*NONE <alphanum-name 1..8>	*NONE <alphanum-name1..40> / <c-string 1..40>	*NONE <c-string 1..64 with-low>
PASSWORD =	*NONE <c-string 1..32> / <x-string 1..16>	*NONE <alphanum-name 1..8>	*NONE <c-string 1..64 with-low> / <x-string 1..128>

USER-IDENTIFICATION =
Identification of the user (user ID) in the remote system.

ACCOUNT =
Account number of the user in the remote system.

ACCOUNT = *NONE
relevant for *BS2000, *MSP and *ANY.
The remote system does not require an account number.

PASSWORD =
Password authorizing the user to access the remote system.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

Follow-up processing in the remote system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING.
The individual commands must be separated by a semicolon. If a character string is enclosed in single or double quotes (’ or ") within a command sequence, openFT will not interpret any semi-colon present within this character string as a separator.

Example

```
SUCC=’command1;command2;command3’
```

The entries for the operands SUCCESS-PROCESSING and FAILURE-PROCESSING may total up to 1000 characters. If the length limit for follow-up processing data is exceeded due to variable replacement, openFT ends the request and returns an error message

SUCCESS-PROCESSING =
Follow-up processing to be executed in the remote system after a successful file transfer.

FTP partners do not support follow-up processing.

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
relevant for:	X	X	X
default value:	*NONE	*NONE	*NONE

SUCCESS-PROCESSING = *NONE
No follow-up processing is to be executed.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
Command to be executed in the remote system after a successful file transfer. It must be specified in quotes according to the syntax and conventions of the remote system.

When in the form of a c-string, follow-up processing is enclosed in quotes.

In the case of a BS2000 partner, each command begins with a slash ('/command'). It is also possible to specify a sequence of commands separated from one another by semicolons.

In the case of z/OS partners (*MSP), the comments relating to follow-up processing in the local system apply, i.e. every command starts without a backslash ('command'). You can also specify a sequence of commands separated by semicolons (see page 336)..

For all other partner systems (*ANY), the rules applying to the remote system in question apply.

When entering the follow-up processing specifications, you may also use variables provided that these are supported by the remote system (see also page 116).

FAILURE-PROCESSING =
Follow-up processing to be executed in the remote system after an unsuccessful file transfer. This follow-up processing is only started if a file transfer that has already commences is terminated due to an error.

FTP partners do not support follow-up processing.

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
relevant for:	X	X	X
default value:	*NONE	*NONE	*NONE

FAILURE-PROCESSING = *NONE
No follow-up processing is to be executed.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
Command to be executed in the remote system if the file transfer is aborted because of an error. It must be specified in quotes according to the syntax and conventions of the remote system.

The same specifications are valid for FAILURE-PROCESSING as for SUCCESS-PROCESSING, see above.

CODED-CHARACTER-SET=
Coding (character set) that is to be used to read or write the remote file.

CODED-CHARACTER-SET= *STD
The character set used by default to read or write the remote file is the character set defined as the default in the remote system.

CODED-CHARACTER-SET= <alphanum-name 1..8> / <c-string 1..8 with-low>
Coding (CCS) that is to be used to read or write the remote file. The character set must be known in the remote system.

Optional entries

The optional entries permit you to set special conditions for the operation and time frame of your file transfers. The optional entries deal with the type of data transfer,

– compressed (COMPRESS) or
– encrypted (DATA-ENCRYPTION),
– specify the coding of the send file (DATA-TYPE),
– set the write rules for the receive file (WRITE-MODE) and
– specify the maximum record length (RECORD-SIZE).
– specify the tabulator expansion (TABULATOR)

COMPRESS =
Defines whether the data in the send file is to be transferred in compressed form.

COMPRESS = *NONE
The data in the send file is transferred uncompressed.

COMPRESS = *BYTE-REPETITION
The data in the send file is transferred in compressed form. Compression affects consecutive bytes with identical contents. If file transfer in compressed form is not possible , the data is transferred in uncompressed form.

COMPRESS = *ZIP
The data in the send file is transferred in compressed form. Compression affects consecutive bytes with identical contents. If file transfer in compressed form is not possible , the data is transferred in uncompressed form.

WRITE-MODE =
Determine how the data is to be written into the receive file. Three options are available. You can

– overwrite an already existing file in the receiving system.
– set up a new file in the receiving system. If a file with the same name already exists in the receiving system, it will not be overwritten.
– attach the transferred file to a file which already exists in the receiving system.

WRITE-MODE = *REPLACE-FILE
Overwrites the receive file from start of file. If the receive system already contains a file with this name, this file and where necessary its file attributes are overwritten. The previous contents of this file are thus completely erased. If the destination does not already exist, it is newly created.

WRITE-MODE = *NEW-FILE
Writes the receive file from start of file. If the receive system already contains a file with this name, this file is not overwritten and the send file is not transferred.

It should be noted that the receive file can already exist following the abortion of a file transfer request. It is not deleted in this case. If a new attempt is made, the request is rejected in the case of WRITE-MODE=*NEW-FILE, as the file already exists.

WRITE-MODE = *EXTEND-FILE
The receive file is extended from the end of file and written to end of file from this point. If the receive system does not yet include a file with this name, a new receive file is created. If the partner is a BS2000 system, then it depends on the system characteristic whether a request with the specification WRITE-MODE=EXTEND-FILE will be accepted or not. The specification WRITE-MODE=*EXTEND-FILE is not permitted when transferring an entire PO or PDSE data set.

The specification WRITE-MODE=EXTEND-FILE is permitted in other cases only if
– send file and receive file have the same record formats,
– for send files and receive files with fixed-length records the record length is the same, and
– the buffer of the receive file can accept the largest record in the send file.

If a file transfer with WRITE-MODE=EXTEND-FILE is aborted permanently, the receive file retains the contents it had at the moment the transfer was terminated.

DATA-TYPE =
Coding used for data in the send file.

DATA-TYPE = *NOT-SPECIFIED
For openFT partners:
The specification is interpreted in the same way as DATA-TYPE=*BINARY if the partner system is an openFT for the BS2000 system and the transferred file is neither a POSIX file nor a library member. Otherwise the specification is interpreted in the same way as DATA-TYPE=*CHARACTER.
For FTAM partners:
The send file type is unknown and is defined by the send system.
In z/OS, the specification is interpreted as DATA-TYPE = *CHARACTER.

DATA-TYPE = *USER
The send file contains structured binary data of variable record length. On Unix and Windows systems, a 2-byte field specifying the record length precedes each record. The maximum record length is 32767 bytes.

DATA-TYPE = *CHARACTER(...)
The send file is transferred as a text file. The receive system stores the file in its character code as text (i.e. a code conversion is performed on the file if necessary).

DATA-TYPE = *BINARY(...)
The send file is transferred as a binary file. The receive system stores the file as it was supplied by the send system. No code conversion takes place.

TRANSPARENT =
Specifies if the file is to be converted to a transparent format.

TRANSPARENT = *NO
No transparent format should be generated. If a file in transparent format is sent to a system that supports transparent transfer, then the file is automatically set up there again with its original attributes.

TRANSPARENT = *YES
This specification is only of use to retrieve files from a partner system that supports transparent transfer. z/OS can then act as temporary storage for such files. The partner system converts the send file into a transparent format and flags it internally as a text or binary file.
If a transparent file is to be returned to the partner, this must always take the form of a binary file with TRANSPARENT=*NO. The partner system automatically recognizes that it has received a transparent file and creates the file again with its original attributes.
In the case of send requests, TRANSPARENT=*YES is ignored.

PRIORITY =
Priority with which the file transfer is initiated relative to other file transfers to the same remote system.

PRIORITY = *NORMAL
The file transfer has normal priority.

PRIORITY = *HIGH
The file transfer has high priority.
Requests with high priority executed via openFT protocols can interrupt normal priority requests for the time it takes to terminate those high priority requests. The interrupted requests are then restarted.

PRIORITY = *LOW
The file transfer has low priority.

START =
Time when the file transfer is to start. The application of the operand is accurate to approximately 5 minutes.

START = *SOON
The file transfer starts as soon as the resources required are available.

START = *EARLIEST(...)
The file transfer starts as soon as the resources required are available and not prior to the time specified. Up to this point the file transfer request is kept in a HOLD state. The date and time specified must not be further ahead than 22 days and 14 hours at the most. If the date and time specified have already passed, the file transfer is executed as if START=*SOON had been specified.

DATE =
Day when the file transfer is to be initiated.

DATE = *TODAY
The file transfer is initiated at the earliest on the day the command is issued.

DATE = *TOMORROW
The file transfer is initiated at the earliest on the day following issue of the command.

DATE = <date 8..10>
The file transfer is initiated on the calendar day specified. If the year is defined by four digits, it must be a year between 1960 and 2059. If only two digits are entered, an internal procedure extends the figure to four digits to denote a year between 1960 and 2059.

TIME = 00:00 / <time 1..8>
The file transfer is initiated at the earliest on the day following issue of the command.

CANCEL =
Specifies whether and when the file transfer is to be aborted. The application of the operand is accurate to approximately 5 minutes.

CANCEL = *NO
The file transfer is not to be deliberately aborted.

CANCEL = *AT(...)
The file transfer is to be aborted at a specific point in time.
The time specified must not
– have already passed,
– be more than 22 days and 14 hours after the specified start time,
– be before or the same as the time specified in the START operand.

DATE =
Day when the file transfer is to be aborted.

DATE = *TODAY
The file transfer is aborted on the day the command is issued.

DATE = *TOMORROW
The file transfer is aborted on the day following issue of the command.

DATE = <date 8..10>
The file transfer is aborted on the calendar day specified. If the year is defined by four digits, it must be a year between 1960 and 2059. If only two digits are entered, an internal procedure extends the figure to four digits to denote a year between 1960 and 2059.

TIME = 23:59 / <time 1..8>
The file transfer is aborted at the specified time on the chosen calendar day.

DATA-ENCRYPTION =
Determines whether or not the file transfer is to be encrypted.

DATA-ENCRYPTION = *NO
The file contents are not transmitted in encrypted form.

DATA-ENCRYPTION = *YES
The file contents are transmitted in encrypted form. If encryption is not an available in the local system, the request is rejected with the error message FTR2111. If the partner system does not permit encryption, the request is rejected with the error message FTR2113.

DATA-ENCRYPTION = *ONLY-DATA-INTEGRITY
The data integrity of the transferred file content is checked using cryptographic means. In the case of openFT partners, this ensures that malevolent attempts to manipulate data during transfer are detected. If an error occurs, openFT performs a restart for asynchronous transfer requests.
If the partner system does not support data integrity checking (e.g. openFT < V8.1), the request is rejected.

In the case of requests with data encryption (*YES), data integrity is also automatically checked. Transfer errors in the network are automatically detected by the checking mechanisms of the transfer protocols used. Data integrity checking is not necessary for this.

RECORD-SIZE =
Maximum record length of the data that is to be transferred. If the record length of the send file is not known from the catalog (e.g. transfer of files from Unix systems, Windows systems or POSIX), the RECORD-SIZE specification is used as the maximum record size. If a record is transferred that exceeds this maximum record size, the request is aborted with

```
FTR2087 OPENFT: Request >>1<<. File structure error >>2<<
```

RECORD-SIZE = *NOT-SPECIFIED
As before. The maximum record length is automatically determined from the catalog.

RECORD-SIZE = <integer 1..32756>
Maximum record length of the data that is to be transferred.

RECORD-FORMAT =
Indicates how the data is transferred on a file transfer to or from a partner.

RECORD-FORMAT = *STD
The record format specification is unchanged.

RECORD-FORMAT = *FIXED
The data is transferred in fixed length records.

RECORD-FORMAT = *VARIABLE
The data is transferred in variable length records.

RECORD-FORMAT = *UNDEFINED
The record length used for data transfer is not mapped to the real system. This means that the record length used for transfer is not identical to the record length in the real file.
In the case of text files, each record is terminated with an end-of-record character both during transfer and then in the real system. Binary files are stored as bit strings in the real system .

TABULATOR =
Specifies whether tab expansion is activated.

TABULATOR = *AUTO
The system uses tab expansion as required.

TABULATOR = *ON
Tab expansion is activated.

TABULATOR = *OFF
Tab expansion is deactivated.

TARGET-FILE-FORMAT =
This operand allows the format of the target file to be specified.

TARGET-FILE-FORMAT = *SAME
The format of the target file is to be the same as that of the send file.

TARGET-FILE-FORMAT = *BLOCK-ORIENTED
The file is to be stored with a block structure. As of openFT V11.0, support is only offered for creating a block-structure file in BS2000 and in PAM format. Creation of a block-structure file in the remote system is only supported with the openFT protocol. Transfer must be performed in binary format. If the file type is specified neither in the command (DATA-TYPE) nor in the file catalog, binary transfer is automatically assumed.

The PAM file created depends on the pubset type (PAMKEY, DATA, DATA-4K). Each of the blocks is completely filled with the binary data stream received. If the data originally comes from a PAM file, the PAM keys are lost during transfer, and the file structure may be lost if the formats of the sending and receiving pubsets differ.

If openFT V10 is running on the receiving system, the file is created as a sequential file with an undefined record format. If older openFT versions are used in the receiving system, the request is rejected.

TARGET-FILE-FORMAT = *SEQUENTIAL (...)
The format of the target file is to be sequential. This also makes it possible to read block-structure files and index sequential files sequentially. The reading of PAM files and ISAM files in BS2000 is supported in openFT version 11.0:

– A PAM file is mapped to a binary sequential file with an undefined record format. The transfer is compatible with standard FTP transfer in BS2000.

– An ISAM file is mapped to the corresponding sequential format (fixed or variable record format). The contents of the ISAM keys is retained in the records, but the key positions are lost.

Specifying *SEQUENTIAL for a sequential send file has no effect.

RECORD-FORMAT =
The record format can be specified for a sequential target file.

RECORD-FORMAT = *SAME
The record format of the target file is to be the same as that of the send file.

RECORD-FORMAT = *UNDEFINED
The record format of the target file is to be undefined. The record structure of the send file is lost. (At least) one block is written for each transfer unit on target systems running BS2000 or z/OS. This can significantly increase the required disk storage space, for instance if the send file is made up of variable length records.

If the FT request is free of errors from the perspective of the local system, then the FT system outputs the following report as an FT request confirmation:

```
FTR0000 OPENFT: Request (&00) accepted
```

(&00) in this case, is the Identification of the FT request that assigns the local FT system to each FT request. Using this FT request ID, you can cancel the FT request (NCANCEL) command), or you can get information on the status of the FT request (NSTATUS command). The FT request ID may consist of a maximum of ten decimals. You can, of course, access your FT requests, even if you do not know the FT request ID (see the information following page 353).

If the local system cannot accept the request, it issues the relevant error message (e.g. due to a syntax error in the command or because the user is not authorized to access the send or receive file). The FT messages are explained in the appendix (page 411ff).

If the request has been accepted by the local system but cannot be executed, you will find the relevant error message in the result list, provided you have specified that a result list is to be created (see the LISTING parameter).

Information on the asynchronous messages issued by the local system on termination of the file transfer request is given in the section “Messages and return codes automatically issued by openFT for z/OS” on page 108

If neither a result list nor asynchronous messages provide information on the success or failure of the request, you can use the logging function to determine whether the request has been completed.

5.30.3 Examples of the NCOPY command

This section provides sample applications of the NCOPY command.

Further examples are available on the openFT product volume; please consult your FT administrator.

1. NCOPY command for openFT with mandatory operands only

   When the conditions on page 313 apply, the NCOPY command can be entered only with the mandatory operands.

   In the following example the file DATA is to be transferred from the local computer to the partner computer HOST001.
   The command is entered in the recommended short form.

   ```
   NCOPY TO,HOST001,(FILE=DATA),*MSP(FILE=DATA,TRANS=(MICKEY))
   ```

   The long form of this command is as follows:

   ```
   NCOPY TRANSFER-DIRECTION=TO,PARTNER=HOST001,
   LOCAL-PARAMETER=(FILE-NAME=DATA),
   REMOTE-PARAMETER=*MSP(FILE-NAME=DATA,
   TRANSFER-ADMISSION=(USER-IDENTIFICATION=MICKEY,ACCOUNT=1313,PASSWORD=abc))
   ```

2. Transfer of a file with password protection cataloged under another user ID in the remote system.

   The file LIST is stored in computer HOST002 under the user ID SHIPPING and protected by the password C'XX'. The ID SHIPPING has the account number SHIP002 and the password TOPSEC.
   The command is entered in the local z/OS system under the identifier CENTRAL.

   The example below shows both the short form and the long form of the command used to transfer the file LIST to the local system and store it there in the file *LIST.ABC*, which has not yet been created. If this file already exists, the LIST file should not be transferred.

   Recommended short form of the command:

```
NCOPY FROM,VAR002,(LIST.ABC),*MSP(LIST,XX, -
          (SHIPPING,SHIP002,TOPSEC)),,NEW

FTR0000 OPENFT: Request 31485389 accepted
```

   Long form of the command:

```
NCOPY PARTNER=HOST002,TRANSFER-DIRECTION=FROM-PARTNER,
LOCAL-PARAMETER=(FILE-NAME=LIST.ABC),
REMOTE-PARAMETER=*MSP(FILE-NAME=LIST,PASSWORD=XX,
TRANSFER-ADMISSION=(USER-IDENTIFICATION=SHIPPING,
ACCOUNT=SHIP002,PASSWORD=TOPSEC)),
WRITE-MODE=*NEW-FILE

FTR0000 OPENFT: Request 16085132 accepted
```

3. Collection of files
   A central office has to collect the monthly reports from its 5 branch offices on the first of every month. These monthly reports are edited ready for printing in each of the branch offices and contained in a file called REPORT.*month* and are each to be transferred into a file in the central location called REPORT.*month.branch-office*. The user IDs do not contain any passwords.

   The transfer of these files is carried out with the following CLIST procedure:

```
PROC 1 MONTH
/*       PLEASE ENTER LAST MONTH FOR MONTH!                          */
NCOPY TRANS=FROM,PARTNER=BRANCH1, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH1, +
SUCC='SEND ''REPORT.&MONTH..BRANCH1 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTRO1))
NCOPY TRANS=FROM,PARTNER=BRANCH2, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH2, +
SUCC='SEND ''REPORT.&MONTH..BRANCH2 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTRO1))
NCOPY TRANS=FROM,PARTNER=BRANCH3,  +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH3,  +
SUCC='SEND ''REPORT.&MONTH..BRANCH3 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTRO1))
NCOPY TRANS=FROM,PARTNER=BRANCH4, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH4, +
SUCC='SEND ''REPORT.&MONTH..BRANCH4 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTRO1))
NCOPY TRANS=FROM,PARTNER=BRANCH5, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH5, +
SUCC='SEND ''REPORT.&MONTH..BRANCH5 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTRO1))
END
```

   This CLIST procedure (name: *MONTH.CLIST*) is called as follows, taking the month of November as an example:

```
EX MONTH 'NOVEMBER'
FTR0000 OPENFT: Request 30436972 accepted
FTR0000 OPENFT: Request 68185709 accepted
FTR0000 OPENFT: Request 38825582 accepted
FTR0000 OPENFT: Request 31485551 accepted
FTR0000 OPENFT: Request 37777008 accepted
```

4. Distribution of files

   A central office distributes guidelines to its five branch offices.
   This guidelines are subsequently printed at the receive system by a job which is contained in the member *PRINT* of the PO file *JOB*.

   To distribute the guidelines the central office uses the the following job:

```
//CENTRAL  JOB
//NCOPY    EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
NCOPY TRANS=TO,PARTNER=BRANCH1,                                    -
LOC=(FILE-NAME=GUIDE.LINE),                                        -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA1),                        -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH2,                                    -
LOC=(FILE-NAME=GUIDE.LINE),                                        -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA2),                        -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH3,                                    -
LOC=(FILE-NAME=GUIDE.LINE),                                        -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA3),                        -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH4,                                    -
LOC=(FILE-NAME=GUIDE.LINE),                                        -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA4),                        -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH5,                                    -
LOC=(FILE-NAME=GUIDE.LINE),                                        -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA5),                        -
SUCC='ALLOC DSNAME(JOB(PRINT))')
/*
//
```

   This job is stored in the member GUIDEL of the PO file SHIPPING.CNTL, which is called up as follows under user ID CENTRAL:

```
SUB SHIPPING(GUIDEL)

JOB CENTRAL(JOB00136) SUBMITTED
```

5. Chaining of files

   A central office collects SAM files (e.g. transaction files) from its 3 branch offices. The files are to be concatenated and are not to be processed until all the files to be collected have been transferred. The files have the name SAM.FILE in the example and are to be stored consecutively in the file SAM.ALL.
   The user IDs do **not** contain any passwords.

   The files are transferred using the following CLIST procedure (name: *CHAIN.CLIST*):

```
PROC 0
NCOPY TRANS=FROM,PARTNER=BRANCH1,WRITE=EXT,                          -
REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA1,TRANS=(BRA1,,)),               -
LOC=(FILE-NAME=SAM.ALL,                                              -
SUCC='NCOPY TRANS=FROM,PARTNER=BRANCH2,WRITE=EXT,                    -
REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA2,TRANS=(BRA2,,)),               -
LOC=(FILE-NAME=SAM.ALL,                                              -
SUCC=''NCOPY TRANS=FROM,PARTNER=BRANCH3,WRITE=EXT,                   -
LOC=(FILE-NAME=SAM.ALL,                                              -
REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA3,TRANS=(BRA3,,))'')')
```

   You call the procedure as follows:

```
EX CHAIN

FTR0000 OPENFT: Request 30436727 accepted
```

6. File transfer between openFT for z/OS and another FT system

   The file FILE is to be transferred for z/OS to another system using openFT. In the other system (SYS) the user ID BROOKLYN with the password 20000 is to be used. The file is to be given the name f/i/l/e in this system. The short form of the command is used:

```
NCOPY TRANS=TO,PARTNER=SYS,LOC=(FIL-NAMEE=FILE),
REM=*ANY(FILE-NAME='f/i/l/e',TRANS=('BROOKLYN','20000'))

FTR0000 OPENFT: Request 15078927 accepted
```

   Note the comments in section “Differentiation between uppercase and lowercase letters” on page 153.

7. File transfer to BS2000 systems

   The file FILEB is to be transferred from an z/OS computer to the BS2000 computer BS2. The file is protected in the local system by password ZZZZ. The user ID has the account number ZENTR01 and is protected by password HQ1. In the BS2000 computer the file is to be called FILE, is to be protected against overwriting with the password C'XXXX' and be stored under the user ID CENTRBS2 with the account number CENTRAL2 and password C'CEN2'. The command is entered in the short form.

   ```
   NCOPY TO,BS2,
   (FILE-NAME=FILEB,PASS=ZZZZ),
   *BS2000(FILE=FILE,PASS=C'XXXX',TRANS=(CENTRBS2,CENTRAL2,C'CEN2'))

   FTR0000 OPENFT: Request 56465400 accepted
   ```

8. File transfer using openFT for Unix systems

   The file mailbag is to be obtained from the Unix system ALFRED and transferred to the local z/OS system. The file is held by ALFRED under the user ID „flyte“ with a password of 144002 and is to be given the name NO.HURRY in the local system. An abbreviated command for this request is as follows:

   ```
   NCOPY FROM,ALFRED,(FILE-NAME=NO.HURRY),
   *ANY(FILE-NAME='mailbag',TRANS=('flyte',,'144 002'))

   FTR0000 OPENFT: Request 19458206 accepted
   ```

   Note the comments in section “Differentiation between uppercase and lowercase letters” on page 153.

   In this example, it is assumed that the file to be transferred is a text file (default value *CHAR for the parameter DATA-TYPE). If you want to transfer structured or unstructured binary files, please refer to the section “Binary transfer with openFT for Unix and Windows systems” on page 90.

9. File transfer using FTAC

   The file TURNOVER is to be transferred to the computer JACKJOHN. On this computer openFT for z/OS is installed along with FTAC functionality for enhanced data protection and access control. An FT profile is provided in the computer JACKJOHN for the file transfer request.

   In order to be able to work with this FT profile, the TRANSFER-ADMISSION ‘FORMY-DEARSTEVEN’ must be specified as transfer admission for the computer JACKJOHN. Specifying this gives direct access to the FT profile in the computer JACKJOHN. This FT profile contains the predetermined name that the file is to receive in the computer JACKJOHN and the predetermined details concerning follow-up processing (SUCCESS-PROCESSING and FAILURE-PROCESSING) in this computer. The value *SAME is defined for the PROCESSING-ADMISSION.

The file transfer request must therefore contain the value *NOT-SPECIFIED for the file name.
This specification corresponds to the default value and can therefore be omitted.
The specification *NONE that is required for SUCCESS-PROCESSING and FAILURE-PROCESSING in this example also corresponds to the default value and can therefore be omitted. The default value *SAME for PROCESSING-ADMISSION is accepted by FTAC even if it is prespecified in the admission profile and can therefore **also** be omitted.

The long form of the command for the file transfer is thus as follows:

```
NCOPY TRANSFER-DIRECTION=TO,                                          -
      PARTNER=JACKJOHN,                                               -
      LOCAL-PARAMETER=(FILE-NAME=SALES)                               -
      REMOTE-PARAMETER=*MSP(TRANSFER-ADMISSION='FORMYDEARSTEVEN')
```

There is of course a short form:

```
NCOPY TO,JACKJOHN,(SALES),                                            -
*MSP(TRANS-AD='FORMYDEARSTEVEN')
```

10. Local file processing between two openFT forz/OS systems

    A list of the names of files for the local ID is to be transferred to the remote file SFA-FILE.LOCAL.

```
NCOPY                                                                 -
   TRANSFER-DIRECTION=*TO-PARTNER, PARTNER=ZOSPART,                   -
   LOCAL-PARAMETER=(FILE-NAME=                                        -
   C'|LISTCAT OFILE(SYSPRINT)',                                       -
   TRANS-ADM=(USER=STEVEN,ACCOUNT=XXXX,PASS=TOPSEC)),                 -
   REMOTE-PARAMETER=*MSP(FILE-NAME=SFA-FILE.LOCAL,                    -
   TRANSFER-ADMISSION=PROFZOSPART)

   FTR0000 OPENFT: Request 197292 accepted
```

11. Remote pre-processing between two openFT for z/OS systems

    A list of the FT partner systems in the remote system is to be transferred to the local file INFO.ZOSPART.

```
NCOPY                                                                 -
   TRANSFER-DIRECTION=*FROM-PARTNER, PARTNER=ZOSPART,                 -
   LOCAL-PARAMETER=(FILE=INFO.ZOSPART),                               -
   REMOTE-PARAMETER=*MSP(FILE=                                        -
   C'|FTSHWPTN OUT=*STDOUT',                                          -
   TRANSFER-ADMISSION=PROFZOSPART)

   FTR0000 OPENFT: Request 197294 accepted
```

12. FTINFO command for remote pre-processing

    You want to determine what openFT version is installed on a remote computer.

```
/TRANSFER-FILE                                                    -
/   TRANSFER-DIRECTION=*FROM-PARTNER, PARTNER=UNKNOWN,            -
/   LOCAL-PARAMETER=(FILE-NAME=FTINFO.UNKNOWN),                       -
/   REMOTE-PARAMETER=*ANY(FILE-NAME=C‘|ftinfo -csv‘,                  -
/   TRANSFER-ADMISSION=C’PROFUNKNOWN’)

    FTR0000 OPENFT: Request 197296 accepted
```

    The file FTINFO.UNKNOWN then has the following content:

```
CmdUiVer;OsType;UserId;IsFtAdm;IsFtacAdm;FtLang
900;"z/OS";"OPFTAAA";1;1;"E"
```

    The output, in sequence of occurrence, has the following meaning:
    openFT V9.0 is installed on the remote system and the operating system is z/OS. FTINFO was issued under the user ID OPFTAAA which possesses both FT and FTAC administrator admissions (otherwise "0" instead of "1"). The openFT user interface in the remote system "speaks" English (otherwise "D" for German).

5.31 NSTATUS
Query status of file transfer request

Note on usage

User group: FT user and FT administrator

Alias name: FTSHWREQ

Functional description

The NSTATUS command allows you to request information about FT requests. As with NCANCEL, you can specify selection criteria in order to obtain information about specific FT requests.

FT users can only obtain information about the FT requests they own.

The owner of requests issued in the local system is the user ID under which they are submitted. The owner of requests issued in the remote system is the user ID in the local system under which the requests are executed.

The scope of information to be output can be selected. By default the following information is output by the system in response to the NSTATUS command:
– the transfer ID of the request,
– the initiator of the request (local or remote system),
– the operating status of the request (see description of operands for more details),
– the partner system,
– the transfer direction,
– the name of the file to be transferred in the local system.
– the number of bytes transferred

By entering INFORMATION=*ALL in the NSTATUS command more information can be obtained. openFT then, in addition to the standard output, outputs the values of further operands of the transfer command that was used to issue the request. Which output parameters are displayed depends on the parameters which were specified for the request.

The complete description of all possible output parameters and values is provided in the section “Meaning of the fields in the long output” on page 361.

The more precise your information request, the fewer irrelevant requests are output.

When you specification of INFORMATION=*SUMMARY returns a small table with the number of jobs in the various request states.

Format

NSTATUS / **FTSHWREQ**

TRANSFER-ID = ***ALL** / <integer 1..2147483647>

,SELECT = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

OWNER-IDENTIFICATION = ***OWN** / <name 1..8>

,INITIATOR = (***LOC**AL, ***REMOTE**) / list-poss(2): ***LOC**AL / ***REM**OTE

,PARTNER = ***ALL**(...) / <text 1..200 with-low>

***ALL**(...)

PARTNER-STATE = ***ALL** / ***ACTIVE**

,FILE-NAME = ***ALL** / <filename 1..59> / <c-string 1..512 with-low>

,MONJV = ***NONE** /

,JV-PASSWORD = ***NONE**

,STATE = ***ALL** / ***SUSP**END / ***LOCK**ED / ***WAIT** / ***ACT**IVE / ***CAN**CELLED / ***FIN**ISHED / ***HOLD**

,**GLOB**AL-**REQU**EST-**ID** = ***ALL** / <alphanum-name 1..10>

,INFORMATION = ***STD** / ***ALL** / ***SUMMARY**

,OUTPUT = ***STDERR**(...) / ***STDOUT**(...)

***STDERR**(...) / ***STDOUT**(...)

LAYOUT = ***STD** / ***CSV**

Operands

TRANSFER-ID =
Transfer ID of the FT request about which information is required.

TRANSFER-ID = *ALL
Supplies information about all the owner's FT requests.
The FT user can only obtain information about the current requests he/she owns.

TRANSFER-ID = <integer 1..2147483647>
Transfer ID assigned to the local system and output as part of the message confirming acceptance of the request.

SELECT =
Contains selection criteria defining the file transfer requests on which inquiries are to be made. Information on a file transfer request is output if the request satisfies all the specified criteria.

SELECT = *OWN
Provides information on all current file transfer requests for which you are designated as the owner.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
Owner of the FT requests.

OWNER-IDENTIFICATION = *OWN
Provides information only on the file transfer requests in the user's own ID.

OWNER-IDENTIFICATION = <name 1..8>
Specific user ID about whose file transfer requests information is required. The FT user may only enter his/her own user ID. The specification corresponds to *OWN.

INITIATOR =
Initiator of the file transfer requests concerned.

INITIATOR = (*LOCAL,*REMOTE)
Provides information on file transfer requests in the local system and in remote systems.

INITIATOR = *LOCAL
Provides information on file transfer requests issued in the local system.

INITIATOR = *REMOTE
Provides information on file transfer requests issued in the remote systems.

PARTNER =
Selects file transfer requests carried out with a specified remote system.

PARTNER = *ALL(...)
The partner system is not used as a selection criterion to determine the file transfer requests on which information is to be output.

PARTNER-STATE =
The status of the partner system is used as a selection criterion.

PARTNER-STATE = *ALL
The requests are selected independently of the partner system's status.

PARTNER-STATE = *ACTIVE
Only the requests to and from the active partners are selected.

PARTNER = <text 1..200 with-low>
Name or an address of a partner system. Information is required on the file transfer requests being executed with this system. For more information on address specifications, see section "Defining the partner computer" on page 99.

FILE-NAME =
FT requests that access this file in the local system as a send file or receive file. The file name or library member name must be specified exactly as it appears in the FT request. If %UNIQUE was specified, the file name generated by openFT must be entered as the selection criterion here.

FILE-NAME = *ALL
The file name is not used as a selection criterion to define the file transfer requests on which information is to be output.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Name of a file. Information is required on the file transfer requests that access this file.

MONJV = *NONE
The parameter is supported for reasons of compatibility only.

JV-PASSWORD = *NONE
The parameter is supported for reasons of compatibility only.

STATE =
Selects those file transfer requests that are in the specified status. The status of a request may change in between entry of the command and information output. This is why the output may include requests that are in a state other than the one selected with STATE.

STATE = *ALL
The status of a request is not used as a selection criterion to define the file transfer requests on which information is to be output.

STATE = *SUSPEND
Requests information on those file transfer requests that are currently in SUSPEND status (= interrupted).

STATE = *LOCKED
Requests information on FT requests that are in the LOCKED operating status (= temporarily locked as a result of a longer term resource bottleneck).

STATE = *WAIT
Requests information on those file transfer requests that are currently in WAIT status (= waiting for resources).

STATE = *ACTIVE
Requests information on those file transfer requests that are currently in ACTIVE status (= being processed).

STATE = *CANCELLED
Requests information on those file transfer requests that were canceled and are waiting for negotiation with the communications partner to be concluded. These requests are visible only to the FT administrator.

STATE = *FINISHED
Requests information on those file transfer requests that are currently in FINISHED status (= terminated or aborted, but where the user has not yet been informed).

STATE = *HOLD
Requests information on those FT requests that are currently in HOLD status (= awaiting the specified start time).

GLOBAL-REQUEST-ID =
Selects the FT requests on the basis of the global request identification.

GLOBAL-REQUEST-ID = *ALL
The global request identification is not a search criterion.

GLOBAL-REQUEST-ID = <alphanum-name 1..10>
Requests information on the FT request with a particular global request identification. The global request identification is relevant only for inbound requests of openFTpartners. It is assigned by the initiator of the request (transfer ID) and transferred to the local system.

INFORMATION =
Scope of the output.

INFORMATION = *STD
Output is summary form and contains the following information (see “Description of the short output” on page 359):
– Transfer ID,
– Initiator,
– State of the request,
– Partner,
– Direction of transfer,
– Byte count,
– File or library member name in the local system.

INFORMATION = *ALL
Output is in full form. In addition to the summary form data, further information is provided on the operands used in the NCOPY command (see “Description of the long output” on page 360).

INFORMATION = *SUMMARY
Output is in the form of a specified sum. By specifying INFORMATION=*SUMMARY, you can restrict the output information to a statistic of the currently existing requests. By doing this, the display is arranged according to the conditions in which the requests find themselves. The displayed sum can, of course, exceed the sum of the individual columns, since all requests, even those that still have no request condition, are counted. Information is output about the number of request in each individual processing status (see “Description of the summary output” on page 364).

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon ";" (see section "Output in CSV format" on page 164).

If selection criteria are specified in the NSTATUS command and no request is found that matches all the specified criteria, the command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria
```

5.31.1 Description of the short output

Example 1

Information is to be output to SYSOUT on those FT requests submitted by the remote system ALFRED which require access to the file DRAISINE and are currently active. The required command is as follows:

```
NSTATUS SELECT=(INITIATOR=*REMOTE,PARTNER=ALFRED, -
FILE=DRAISINE,STATE=*ACTIVE)
```

The recommended short form of this command is as follows:

```
NSTATUS SEL=(INIT=*REM,PART-NAME=ALFRED,FILE=DRAISINE,STATE=*ACT)
```

The information is then output in the following format, for example:

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
528184     REM ACT   ALFRED   TO   14760       DRAISINE
```

Description of the output columns:

`TRANS-ID:` Transfer ID of the file transfer request

`INI:` Initiator of the file transfer request : *REM* for REMOTE, *LOC* for LOCAL

`STATE:` State of the request (here *ACT* for ACTIVE, other outputs:

`SUSP` for SUSPEND,
Inbound request suspended, e.g. due to higher priority requests.

`LOCK` for LOCKED,

`WAIT` for WAIT,

`FIN` for FINISHED,

`HOLD` for HOLD

`PARTNER:` Symbolic name of the relevant partner system.

If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

* The FT administrator of the local system has locked a resource.

! An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections, because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
The cause can be queried with the FT administrator.

`DIR:` Transfer direction

`BYTE-COUNT:` Number of bytes transferred up to the last restart point (in the case of data compression this is the a number of bytes of compressed data)

`FILE-NAME:` Name of the relevant file or library member in the local system

5.31.2 Description of the long output

The long output is described using an example of an outbound request and an example of an inbound request.

Example 1 (Outbound request)

Full information is to be output to 67054 to SYSPRINT via the FT request with transfer ID . If the file transfer request was issued under the same user ID as that under which the inquiry is made, then the command is as follows:

```
NSTATUS TRANSFER-ID=67054,INFORMATION=*ALL,OUTPUT=*STDOUT
```

The recommended short form of this command is as follows:

```
NSTATUS TRANS=67054,INF=*ALL,OUT=*STDOUT
```

The information output on SYSLST then has the following format, for example:

```
TRANSFER-ID =67054       STORE  =12-07-11 14:37:18  FILESIZE=2000
   STATE    =WAIT        BYTECNT=0
   INITIATOR=LOCAL       TRANS  =TO                 PRIO    =NORM
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR
   TRANSP   =NO          ENCRYPT=NO
   OWNER    =OPFTUID     DICHECK=NO
   PARTNER  =BS2PART
   PARTNER-STATE =ACT
   PARTNER-PRIO  =LOW
   LOC: FILE     =FILE.TEST
        TRANS-ADM=(OPFTUID,ACCOUNT)
        ASYN-MSG =ALL
   REM: FILE     =TEST2
        TRANS-ADM=REMOTE-PROFILE
```

Example 2 (Inbound request)

Full information is to be output 67056 to SYSPRINT on the FT request with transfer ID . If the file transfer request was issued under the same user ID as that under which the inquiry is made, then the command is as follows:

```
NSTATUS TRANSFER-ID=67056,INFORMATION=*ALL,OUTPUT=*STDOUT

TRANSFER-ID =67056       STORE  =12-07-11 14:40:53  FILESIZE=40960000
   STATE    =WAIT        BYTECNT=10372320
   INITIATOR=REMOTE      TRANS  =FROM               PRIO    =
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR               GLOB-ID =721214
   TRANSP   =NO          ENCRYPT=NO                 TABEXP  =NO
   OWNER    =OPFTUID     DICHECK=NO                 RECFORM =VARIABLE
   PARTNER  =BS2PART
   PARTNER-STATE =ACT
   PARTNER-PRIO  =NORM
   FILE     =TEST3
   TRANS-ADM=LAST
```

Meaning of the fields in the long output

The list below describes all fields which can occur in the long output (according to lines). Which fields are output in each particular case depends on the type and the parameters of the request.

`TRANSFER-ID:` Transfer ID of the request

`STORE:` The time at which the request was entered in the request queue

`FILESIZE:` The size of the file in bytes. If the output is flagged with a "K" on the right, the output is in kilobytes. If the output is flagged with "M", the output is in megabytes. The size is only shown here if the request has already been active. In the case of receive requests, a value is only shown here if the partner also sends that value.

`STATE:` State of the request

`BYTECNT:` Number of bytes transferred up to the last restart form (in the case of data compression in compressed form)

`INITIATOR:` Initiator of the request

`TRANS:` Transfer direction, as seen from local system

`PRIO:` Priority with which the request is to be started;
here: `NORM` for NORMAL.

`WRITE:` Specifies if or when the receive file is to be overwritten or extended

START: Requested start time of the request
(SOON for “as soon as possible”)

CANCEL: Requested abortion time
(NO for “no abortion requested”)

COMPRESS: Specifies whether or not the file is to be transferred in compressed form

DATA: Type of file:

- CHAR for text file
- BIN for binary file
- NOT-SPECIFIED
  in TRANSFER-FILE (NCOPY), no DATA-TYPE was specified
- USER for user format

GLOB-ID: Global request identification, displayed only in the case of inbound requests from openFTpartners (INITIATOR=REMOTE). This corresponds to the request identification (=TRANSFER-ID) on the initiator system.

TRANSP: Specifies whether the transfer is to be done in transparent format

ENCRYPT: Specifies whether the file content is to be transferred in encrypted form

TARGFORM: Format of the transferred file in the target system:

- SEQ Sequential file format
- BLOCK Block format

TRECFRM: Record format of the file in the target system:

- STD The same record format as in the sending system
- UNDEFINED
  Undefined record format

OWNER: Owner of request in local system

DICHECK: Specifies whether data integrity is to be checked (YES) or not (NO)

PARTNER: Symbolic name of partner system participating in the request.
If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

* The FT administrator of the local system has locked a resource.

! An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections, because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
The cause can be queried with the FT administrator.

PARTNER-STATE:
Status of the partner. Possible values:

ACT Activated

DEACT Deactivated

NOCON No connection, for instance because the openFT server has not been started on the remote system.

INSTERR
There is an installation or configuration error (for example, the local system is not known to the partner or the address of the partner in the partner list is not valid) or authentication of one of the partners has failed or encryption is not available locally or on the partner system.

PARTNER-PRIO:
Prioritization of the partner when processing requests.
Possible values:

LOW The partner has low priority.

NORM The partner has normal priority.

HIGH The partner has high priority.

LOC: Specifications on the local system (LOCAL-PARAMETER).

The entry can include more than in this example; the keywords correspond to the recommended abbreviations of the keywords of the transfer command; the meaning of the operand is also to be found there.

FILE: Local file name

ASYN-MSG:
Specifies which request result leads to an asynchronous termination message. Possible values: ALL, FAIL.

`REM:` Specifications on the remote system (REMOTE-PARAMETER).

The entry can include more than in this example; the keywords correspond to the recommended abbreviations of the keywords of the transfer command; the meaning of the operand is also to be found there.

`FILE:` Remote file name

The following parameters are only output for locally issued requests.

`TRANS-ADM:`
Transfer admission (here for the remote system. Instead of the triplet user ID, account number and password where appropriate, REMOTE-PROFILE can also be output here if a remote FTAC FT profile is addressed. The equivalent also applies to entries in the local system.

`CCSN:` CCS name used in the local and/or remote system when reading the file.

5.31.3 Description of the summary output

You want to output information about the number of request in each individual processing status.

```
NSTATUS INF=*SUMMARY
 ACT    WAIT    LOCK    SUSP   HOLD    FIN     TOTAL
   3       5       0       0      0      0        10
```

There are three requests in the ACTIVE condition, and five in the WAIT condition. Two requests are still in protocol handling, therefore the sum is 10.

6 Program interface for the FT user

A file transfer request for an FT system can also be submitted from an application program. This function uses the OPENFT macro .

Description of the functions

This program interface has the following functions:

– Calling the openFT commands (with the exception of FTHELP) with all the parameters which are also valid at the command interface (see previous chapter).

– In order to call one of the above-mentioned commands the application uses the ASSEMBLER macro OPENFT. Via the LINK macro, FTNC calls the module NCOPY, which in turn sets up a connection to openFT, sends the command unchanged to openFT and waits for a message from it. After the message has been received from openFT, the connection is cleared down again.

– Messages generated by this module or created by openFT are stored as text strings in a buffer to be provided by the application program. For details of the format of these messages, see the section “FT system messages” on page 411 in the appendix.

  The application program then continues. A return code is supplied to the application program in register 15.

– Messages from openFT are always passed to the application program in a buffer. Even if the application program is running under TSO, no output is made to the terminal.

– With the aid of the LINK macro a search is made for the module NCOPY in the normal hierarchy of libraries in an IBM environment (TASKLIB, JOBLIB, link library). This module and the macro OPENFT are not designed to be reentrant. They can only be run in the address space below 16 Mb, but can process addresses below and above 16Mb.

Description of the interface

The ASSEMBLER macro OPENFT is provided for calling commands.

If an instance other than STD is to be used or if an SVC number other than 211 is used for the openFT subsystem, then the file<openft qualifier>.<inst>.CONN must be assigned at runtime via the DD name OPENFT. This file contains the connection data for openFT. The specifications in angle brackets stand for OPENFT QUALIFIER and the instance name of the instance that is to be used.

The following linkage conventions apply to these calls:

– In an IBM environment, registers 0, 1, 13, 14 and 15 have specific meanings and should not be used by the application program in any other way.

– Registers 0 and 1 are used by the above-mentioned macro for passing parameters.

– In register 13 the application program must pass the address of a save area of 72 bytes, justified on a full word boundary. This save area must lie in the address space below 16 Mb.

– Registers 14 and 15 are used by the above-mentioned macro for calling the module NCOPY. This module is loaded into the address space below 16 Mb.

– When control is returned to the application program, register 15 contains a return code.

The commands to be executed by openFT are passed as "command strings", which must have exactly the same format as if the commands were being entered at the terminal (see the chapter “Command interface” on page 147). The commands must be entered in uppercase letters.

Messages returned by openFT are always passed to the application program as text strings in a buffer provided by the application program. For details of the format of these messages see the section “FT system messages” on page 411 in the appendix.

6.1 Macro OPENFT to call a user command

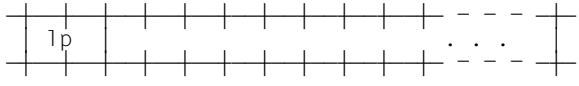
```
[ label ] OPENFT    CMD={cmd-buff-addr}  ,MSG={msg-buffer}  ,VERS={1}
                        {(r)           }       {(r)       }       {2}
```

label
Entry of a symbolic address for the first command in the macro expansion (optional).

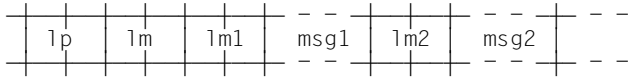
cmd-buff-addr
Address of a reference list which, in turn, also contains addresses which point, amongst others, to the command string. This list must commence on a full word boundary. Its detailed structure is described below.

Instead of a symbolic address it is possible to specify, enclosed in brackets, the number of a register which contains this address. Registers 2 through 12 may be used for this purpose.

msg-buffer
Address of a buffer for the messages or for information passed back to the program. This buffer must commence on a full word boundary and must have the following structure when the macro OPENFT is called:



After the macro OPENFT has been executed this buffer has the following contents:



lp — Length field of 2 bytes. Contains the length of the buffer in bytes (including the length field)

lm — Length field of 2 bytes. Before a command is executed, this field must contain the value X'0000'. After the execution of a command, this field contains the length of the returned text strings in bytes, excluding the lengths of the length fields lp and lm, but including the lengths of length fields lm1, lm2, etc. If the total length of the text strings to be delivered is greater than the buffer provided, then firstly a return code of '10' (hexadecimal) is passed back to the program and the field "lm" contains the length in bytes necessary to accommodate the full text.

lm1 — Length field of 2 bytes. Contains the length of the message text msg1 in bytes, excluding the length of this length field.

msg1 This field contains the first line of the message text output to the terminal as a result of executing the command.

lm2 Length field of 2 bytes. Contains the length of the message text msg2 in bytes, excluding the length of this length field.

msg2 This field contains the second line of the message text output to the terminal as a result of executing the command.

In this way all the lines of the message text are output to the buffer, until either the message text has been output in full or until the buffer length lp has been reached. The last line output to the buffer is then possibly incomplete. Although in this instance no further lines are output to the buffer, the value for "lm" continues to be accumulated, so that eventually lm contains the size which would need to be defined as the buffer size (lp) in order to accept the output in full (excluding the length of the two length fields lp and lm).

The size of the output buffer should be adapted to approximately the volume of expected data. 200 bytes are sufficient to issue an NCOPY while approximately 8KB are required for an FTSHWLOG with 100 logging records.

The format of the messages is identical to those output on the terminal, i.e. if necessary, they contain control characters. These messages are described in the section “FT system messages” on page 411 in the appendix.

Instead of a symbolic address it is possible to specify, in brackets, the number of a register that contains this address. Registers 2 through 12 may be used for this purpose.

VERS=

The return codes have been changed in V10.0 of openFT. This means that different message numbers are output compared with previously (openFT ≤ V9.0). The VERS parameter is used to maintain compatibility with older programs:

1 The old return codes are output, default.

2 The new return codes are output.

The VERS parameter allows you to choose between old and new return codes for output:

– If you require the old return codes to be output (default), compatibility with older programs is maintained. This means that they can be used unchanged and do not even need to be recompiled.

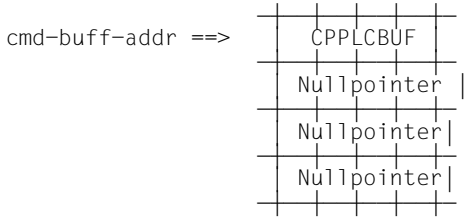
– If you wish to use the new return codes, you must set the parameter VERS=2 explicitly, adapt the programs and recompile them.

Structure of the reference list (cmd-buff-addr)

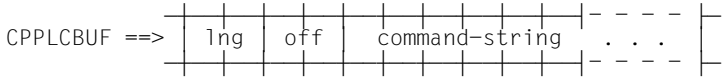
The parameter cmd-buff-addr points to a list, the structure of which is described below.

Transfer of a user command

In the case where a user command is transferred via the macro OPENFT, the parameter cmd-buff-addr contains the address of a list which is four full words in length and corresponds to the "command processor parameter list" (CPPL) in TSO:



This field CPPLCBUF contains the address of the command buffer. The command buffer to which the address in field CPPLCBUF points must commence on a full word boundary and have exactly the same structure as for a TSO command processor.



lng Length field of 2 bytes. This field contains the length of the command string plus the length of fields lng and off (thus, command string + 4).

off Offset field of 2 bytes. This field contains the offset of the first parameter relative to the start of the command string (not relative to the start of this buffer).

command string
This field contains the command to be executed; it is held as a character string. The length of this string is limited to a maximum of 2000 characters. The commands are described in the chapter “Command interface” on page 147.

A return code is sent to the application program in register 15 (right-aligned). Return code (hexadecimal) in R15:

'00' Normal termination

'04' Warning

'0C' Syntax error, access conditions invalid

'10' The buffer was too small for the return message, the text was truncated.

'12' The resources (e.g. VTAM connection) are occupied.

Example of calling a user command

The following NCOPY command is to be executed (using the new return codes):

```
NCOPY  TRANS=TO,PARTNER=BS2,LOC=(FILE=DAT,TRANS=(BERT,A1234,PASSWORD)), -
       REM=*BS2000(INFO,TRANS=(FRED,4567ABC,COVERT))
```

The following ASSEMBLER code is required to effect it:

```
          LA    13,SAVAREA           SET UP SAVE AREA
          OPENFT CMD=CMDBUFF,MSG=MSGBUFF,VERS=2
          .
          .
CMDBUFF   DC    A(CBUFF)             ADDR OF COMMAND BUFFER
          DC    A(0)                 NOT USED
          DC    A(0)                 NOT USED
          DC    A(0)                 NOT USED
          SPACE
CBUFF     DS    0F                   COMMAND BUFFER
          DC    Y(CLNG)              LENGTH FIELD
          DC    Y(L'CNAME)           OFFSET TO 1ST. PARAMETER
CNAME     DC    C'NCOPY   '          COMMAND NAME
          DC    C'TRANS=TO'          1ST. PARAMETER
          DC    C',PARTNER=BS2'      2ND. PARAMETER
          DC    C',LOC=('            LOCAL PARAMETER
          DC    C'FILE=DAT'          FILE NAME
          DC    C',TRANS=('
          DC    C'BERT,A1234,PASSWORD'  TRANSFER ADMISSION
          DC    C')),'
          DC    C'REM=*BS2000('      REMOTE PARAMETER
          DC    C'INFO'              FILE NAME
          DC    C',TRANS=('
          DC    C'FRED,4567ABC,''COVERT'''  TRANSFER ADM.
          DC    C'))'
CBFFEND   EQU   *
CLNG      EQU   CBFFEND-CBUFF        BUFFER LENGTH
          SPACE
MSGBUFF   DS    0F                   MESSAGE BUFFER
          DC    Y(MSGLNG)            BUFFER LENGTH
          DC    X'0000'
          DS    CL200                MESSAGE TEXT
MSGEND    EQU   *
MSGLNG    EQU   MSGEND-MSGBUFF
          SPACE
SAVAREA   DC    18F'0'               SAVE AREA
```

7 What to do if ...

Error FTR2047 directly after NCOPY

Possible cause	Suggested solution
Defective transfer admission in the local system	Specify a valid transfer admission
If openFT-AC is used:	
The transfer admission in the local system does not constitute an authorization for the desired request	Output the return code (RC) of the FTAC logging record with FTSHWLOG␣(RECORD-TYPE=(FT=NONE)),NUMBER=n to discover the reason for rejection with FTHELP xxxx (xxxx=RC)

Error FTR2169 in other cases

Possible cause	Suggested solution
Defective transfer admission in the remote system	Specify a valid transfer admission
The transfer admission in the remote system does not constitute an authorization for the desired request	Determine the reason for rejection in the remote system; in the case of z/OS partners: FTSHWLOG␣(RECORD-TYPE=(FT=NONE)),NUMBER=n and FTHELP xxxx in the case of BS2000 partners: /SHOW-FT-LOGGING-RECORDS␣(RECORD-TYPE=(FT=NONE)),NUMBER=n and /HELP-MSG-INFORMATION␣MSG-IDENTIFICATION=FT-Cxxxx,LANGUAGE=D (xxxx=RC from logging record) in the case of Unix partners: ftshwl␣-rt=c and fthelp␣xxxx (xxxx=RC from logging record) in the case of WINDOWS partners: Logging window and status line outputs RC.

Other FTRxxxx directly after NCOPY

Possible cause	Suggested solution
Error in the local system	Refer to manual

Other FTRxxxx in other cases

Possible cause	Suggested solution
Error in the remote system (in most cases) Error in the local system (less frequently) Exception: FTR0035 `File locked to prevent multiple access.` It is equally likely to be the local or remote file that is locked	Refer to manual

Follow-up processing was not performed

Possible cause	Suggested solution
– Error in the follow-up processing commands and the follow-up processing log was suppressed because the error was intercepted. – Space after a semicolon if multiple follow-up processing commands were issued	Check the follow-up processing commands or do not intercept the error in order to obtain a log from which the error can be identified.
Follow-up processing is required to access files that do not exist or that cannot be accessed with the PROCESSING-ADMISSION	Check the existence of the required files and their access control settings, if necessary correct PROCESSING-ADMISSION
JES rejects follow-up processing because a required "account number" was not specified in the PROCESSING-ADMISSION	Enter the "account number" in the PROCESSING-ADMISSION
The instance-specific job TSOJOB or JCLJOB is incorrectly structured	Ask the FT administrator if there are fundamental problems with follow-up processing

Problems during the execution of preprocessing or postprocessing commands and ftexec calls

Possible cause	Suggested solution
Error in the preprocessing or postprocessing commands or space after a semicolon if multiple preprocessing or postprocessing commands were issued.	Check the preprocessing or postprocessing commands. The job listing may provide additional information concerning the problem.
The instance-specific jobs TSOVVJOB or TSONVJOB are incorrectly structured	Ask the FT administrator if there are fundamental problems with preprocessing or postprocessing

Remote follow-up processing is not performed if an error occurs (FAILURE PROCESSING)

Possible cause	Suggested solution
NCOPY was accepted (FTR0000). However, an error was discovered before the start of file transfer (e.g. receive file locked)	

No result message is output at the terminal

Possible cause	Suggested solution
File transfer not yet terminated	Use NSTATUS to obtain information on the FT request
The initiator is not identical to the user ID in the TRANSFER-ADMISSION	Log on under the user ID specified in the TRANSFER-ADMISSION or ask its user whether a message has been output
No entry after NCOPY	Make an entry at the system
Message not seen or a LOGOFF/LOGON was entered after NCOPY	View logging records
The user has disabled message output with PROFILE NOINTERCOM	Output the message with LISTBC MAIL; for subsequent requests enable message output with PROFILE INTERCOM
	Wait

A deleted request continues to be present in the request file

Possible cause	Suggested solution
The request was already active when the connection to the remote system was interrupted	Wait until the connection is re-established so that openFT can communicate with the partner system concerning the interruption. If the request des not ‘disappear’ after re-establishment of the connection then it can be definitively deleted by the FT administrator using NCANCEL with FORCE-CANCELATION=*YES.

The request is run despite NCANCEL

Possible cause	Suggested solution
NCANCEL came too late. The request had already been completed	

A very large file cannot be transferred

Possible cause	Suggested solution
There is not enough contiguous storage space on the DASD volume	Speak to the system administrator

No information about NCOPY request

Possible cause	Suggested solution
	View with FTSHWLOG␣NUMBER=n

No result list is generated

Possible cause	Suggested solution
LIST=*NONE specified (default value)	
Output lost	
The instance-specific job PRTJOB is incorrectly structured	Ask the FT administrator if there are fundamental problems with result lists

Request remains in WAIT status and flagged with *

Possible cause	Suggested solution
Partner deactivated by FT administrator	Inform system administrator if necessary

Request remains in WAIT status and flagged with !

Possible cause	Suggested solution
Establishment of connection failed:	
Remote system not active	Speak to system administrator of the remote computer
FT in remote system not active	Speak to system administrator of the remote computer
The BCIN/BCACT for your own system is not present in a remote BS2000 system	Speak to system administrator of the remote computer
Maximum number of connections permitted in the remote system currently exhausted	

Request remains in WAIT status and flagged with ?

Possible cause	Suggested solution
The local system is not entered in the remote FT system	Speak to system administrator of the remote computer
No valid partner key is stored in the local system (STATE RAUTH in FTSHWPTN)	The FT administrator must store a current public key for the partner in the local instance's SYSKEY library (Name: symbolic name of the partner system as in the partner list)
No valid local system key is stored in the remote system (STATE LAUTH in FTSHWPTN)	The local FT administrator must send a current public key to the partner system's FT administrator who must store this key in a suitable location

Request remains in WAIT status and is unflagged

Possible cause	Suggested solution
Normal wait for system resources	Wait

7.1 Frequently asked questions

What is the shortest form of the NCOPY command?

The following command is usually sufficient to send a file to a partner system. The items you need to specify yourself start with lowercase characters:

```
NCOPY TO,partner,(file),(,,transAd)
```

For transAd, you can enter an FTAC transfer admission defined in the remote system (e.g. TRANSADM). Alternative input: (user,acc,passwd).

You can also use the same input for Unix partners if you do not mind the fact that the file name is written in uppercase letters there.

The input also functions for PC partners provided that the file name is syntactically permissible there.

Please note: In procedures, you should generally use only guaranteed abbreviations in order to remain independent of the current FT version (for example, *ANY instead of A).

How can I find out which FT requests have been successful and which have not?

If you call the logging records with:

```
FTSHWLOG
```

the result of the last transfer is displayed.

If you want to view the last n entries:

```
FTSHWLOG ,n
```

The most recent entry is displayed at the top of the list.

You can also select on the basis of a wide range of criteria (partner, file name etc.). You should note that when openFT-AC is used, there are two entries for each NCOPY request. First the FTAC entry, identified by a C in the first column, followed by the entry that contains the transfer result (indicated by T).

If you only want to view the transfer results:

```
FTSHWLOG (REC-TYPE=(,N)),n
```

How can I tell whether an error is located in the local or the remote system?

The following rules apply:

If the NCOPY command is not accepted with FTR0000 but is immediately rejected, the error is always located in the local system.

In the case of NCOPY commands that are rejected after being accepted with FTR0000, the error is almost always located in the remote system. If the reason for rejection is
`FTR2169 Remote system: Transfer admission invalid`
then the cause is always located in the remote system.

Cases in which the partner cannot be accessed (e.g. FTR0108) are ambiguous; here it is not generally possible to say where the error lies.

Why is my transfer rejected even though the transfer admission is correct?

It is indeed possible that your request may be rejected even though the transfer admission (e.g. in the form (user,account,password) or TRANSADM has been specified correctly. Your request is also rejected if the transfer admission does not permit all the actions that you want to perform. Some possibilities:

The ID is locked in the remote system (e.g. by SEVER/LOCK-USER in BS2000).

The remote system does not permit any requests that make use of transfer admissions of the form (user, account,password) since all the levels in the FTAC admission set are set to 0.

The required transfer direction or your system are rejected by the partner.

The partner does not permit the required function, e.g. follow-up processing or file management.

Moreover, and particularly when the transfer admission is transferred telephonically, it frequently happens that cases are inverted. Uppercase characters must be specified in quotes if they are to be effective.

Finally, it is also possible that the specified transfer admission was actually incorrect.

My call is rejected with FTR2169 Remote system: Transfer admission invalid. How can I find out why?

You have entered an NCOPY command that was accepted with FTR0000. It was then aborted with FTR2169.

This type of rejection always comes from the partner system. Consequently, the cause can only be determined there.

In the case of openFT products, the reason can be identified very simply by means of the FTAC logging record.

To this end, your partner allows you to view the last logging record or the last n logging records under the receive ID:

in z/OS with `FTSHWLOG [,n]`

in BS2000 with `/SHOW-FT-LOG [,n]`

in Unix systems and under Windows with ftshwl [-nb=n] or via the associated graphical user interface.

You can search for the appropriate FTAC entry (type C or FTAC) on the basis of the partner, file name, time etc. The reason for the rejection is output in the RC column. The meaning of the RCs is directly output at the PC. In BS2000, it can be output with /HELP FTCnnnn and in the other systems with fthelp nnnn (nnnn is the RC).

If your partner cannot find any logging record corresponding to your request then you have either not addressed the right partner or the specified transfer admission does not belong to the expected receive ID. In particular, this will be the case if the transfer admission does not exist (e.g. because you have made a typing error).

What is an FT or FTAC transfer admission and how can I set one up?

In a remote system, you usually identify yourself by means of the logon specifications user ID, account number and password: TRANS-ADM=(user-id,account,password).

It is simpler to use a special admission for file transfer only (TRANS-ADM=transAdm). The owner of the user ID then no longer has to disclose all his or her logon admissions. Instead, the user sets up a so-called admission profile as follows:

in z/OS: `FTCREPRF name,,transAdm`

in BS2000: `/CREATE-FT-PROFILE name,,transAdm`

in Unix systems and under Windows with ftcrep name transAdm or via the associated graphical user interface: menu sequence *File / New / Admission Profile*

name is the name under which the profile is administered (e.g. is deleted again), max. eight-character. transAdm is the admission assigned by the partner and which you specify in your FT command: minimum eight-character. If the specification contains spaces and other special characters or if it is necessary to distinguish between uppercase and lowercase characters, it may be necessary to enclose it in quotes.

Transfer admissions can be set up on systems with openFT-AC.

7.2 Reporting errors

If, despite taking every precaution, an error occurs that you are unable to resolve with the help of the above hints or by following the action suggested for the error message involved, please consult the FT administrator. You will make it easier for the FT administrator to perform diagnostics by providing the following documentation, where applicable:

– a precise description of the circumstances in which the error occurred and whether it can be reproduced,

– a printout of the command in which the error occurred,

– a printout of the error messages which were issued,

– if available, the result list for the request in which the error occurred (if necessary, repeat the request for the purpose of obtaining a result list, see the LISTING parameter in the NCOPY command).

8 Appendix

8.1 Structure of CSV outputs

8.1.1 Output format

The output format for all commands corresponds to the following rules:

– Each record is output in a separate line. A record contains all the information to be displayed on an object.

– The first line is a header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in the record.** In other words, the order of columns is determined by the order of the field names in the header line.

– Two tables, with their own respective headers, are output sequentially for the command FTSHWENV. If one of the tables is empty, the corresponding header is also dropped.

– Individual fields within an output line are delimited by a semicolon “;”.

The following data types are differentiated in the output:

– Number

  Integer

– String

– String: Since ";" is a metacharacter in the CSV output, any text that contains ";" is enclosed in double quotes ("). Double quotes within a text field are doubled in order to differentiate them from text delimiters. When imported into a program, the doubled quotes are automatically removed and the text delimiters removed. Keywords are output in uppercase with a leading asterisk (*) and are not enclosed in double quotes.

– Date

  The date and time are output in the form yyyy-mm-dd hh:mm:ss. In some cases, only the short form yyyy-mm-dd is ouput, i.e. the date alone.

– Time

The time is output in the form yyyy-mm-dd hh:mm:ss or only hh:mm:ss.

Some of the fields in this command output are irrelevant for openFT for z/OS, but they appear nonetheless for reasons of compatibility with other openFT products (e.g. ElemName, ElemPrefix etc0. in the output of FTSHWPRF).

8.1.2 FTSHW

The following table indicates the CSV output format for file attributes.

The values that are marked by an “x” in the **Std** column are also output if INF=*STD is specified. In the case of INF=*NAMES-ONLY, only the FileName column is output.

The **Parameter** column indicates the name of the output parameter in the case of detailed output, see page 248ff.

Column	Type	Values and Meaning	Parameter	Std
FileName	String	File name or directory name enclosed in double quotes / *NSPEC	FILENAME	x
StorageAc-count	String	Account number enclosed in double quotes / *NSPEC	STORAGE-ACCOUNT	x
CreIdentity	String	Identity of the last user of the file (creator) enclosed in double quotes / *NSPEC	CRE name	x
CreTime	Date	Time at which the file was created / *NSPEC	CRE DATE	
ModIdentity	String	Identity of the last user of the file (modification of file content) enclosed in double quotes / *NSPEC	MOD name	
ModTime	Date	Time at which the file was last modified / *NSPEC	MOD DATE	x
ReaIdentity	String	Identity of the last user of the file (file read access) enclosed in double quotes / *NSPEC	REA name	
ReaTime	Date	Time at which the file was last modified / *NSPEC	REA DATE	
AtmIdentity	String	Identity of the last user of the file (modification of file attributes) enclosed in double quotes / *NSPEC	ATM name	
AtmTime	Date	Time at which the file attributes were last modified / *NSPEC	ATM DATE	
FileType	String	*BIN / *DIR / *TEXT / *NONE / *NSPEC File type	file type	x
CharSet	String	*VISIBLE / *IA5 / *GRAPHIC / *GENERAL / *NONE / *NSPEC Character set for the text file if FileType=*TEXT, in the case of another FileType, this is *NONE or *NSPEC	CHARACTERSET	
RecFormat	String	*VAR / *FIX / *NSIG / *NSPEC Record format	RECORD-FORMAT	
RecSize	Number	1... 65535 / *NSPEC Maximum length of the records	RECORD-SIZE	
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC File avaliability	FILE-AVAILABILITY	

Column	Type	Values and Meaning	Parameter	Std
AccessRights	String	nnnnnnnnnn / *NSPEC Access rights, n = p, x, e, a, c, d, t, v, r, -	ACCESS-RIGHTS	x
FileSize	Number	Current file size in bytes / *NSPEC	FILESIZE	x
MaxFileSize	Number	Maximum file size in bytes / *NSPEC	MAX-FILESIZE	
LegalQualif	String	Legal qualification enclosed in double quotes / *NSPEC	LEGAL-QUALIFICATION	
CcsName	String	Name of the character set / *NSPEC	CCS-NAME	

```
FTSHW ZOSMCH01,FILE3,,TRANSADM,INF=*STD,OUT=*STDOUT(*CSV)

FileName;StorageAccount;CreIdentity;ModTime;FileType;AccessRights;FileSize
"FILE3";*NSPEC;"MISTERX";2012-03-19 12:39:47;*NSPEC;r-pxeacd---;2048
```

8.1.3 FTSHWADS

The following table indicates the CSV output format of an admission set.

The **Parameter** column contains the name of the output parameter during normal output, see page 256.

Column	Type	Values and Meaning	Parameter
UserId	String	User ID, enclosed in double quotes / *STD *STD means default admission set	USER-ID
UserMaxObs	Number	0 ... 100 Maximum user level for OUTBOUND-SEND	MAX. USER LEVELS OBS
UserMaxObsStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxObr	Number	0 ... 100 Maximum user level for OUTBOUND-RECEIVE	MAX. USER LEVELS OBR
UserMaxObrStd	String	*YES / *NO *YES means same value as default admission set[1]	
UserMaxIbs	Number	0 ... 100 Maximum user level for INBOUND-SEND	MAX. USER LEVELS IBS
UserMaxIbsStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbr	Number	0 ... 100 Maximum user level for INBOUND-RECEIVE	MAX. USER LEVELS IBR
UserMaxIbrStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbp	Number	0 ... 100 Maximum user level for INBOUND-PROCESSING	MAX. USER LEVELS IBP
UserMaxIbpStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbf	Number	0 ... 100 Maximum user level for INBOUND-FILE-MANAGEMENT	MAX. USER LEVELS IBF
UserMaxIbfStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxObs	Number	0 ... 100 Maximum level of FTAC administrator for OUTBOUND-SEND	MAX. ADM LEVELS OBS
AdmMaxObsStd	String	*YES / *NO *YES means same value as default admission set [1]	

Column	Type	Values and Meaning	Parameter
AdmMaxObr	Number	0 ... 100 Maximum level of FTAC administrator for OUTBOUND-RECEIVE	MAX. ADM LEVELS OBR
AdmMaxObrStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbs	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-SEND	MAX. ADM LEVELS IBS
AdmMaxIbsStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbr	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-RECEIVE	MAX. ADM LEVELS IBR
AdmMaxIbrStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbp	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-PROCESSING	MAX. ADM LEVELS IBP
AdmMaxIbpStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbf	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-FILE-MANAGEMENT	MAX. ADM LEVELS IBF
AdmMaxIbfStd	String	*YES / *NO *YES means same value as default admission set [1]	
Priv	String	*YES / *NO *YES means admission set of FTAC administrator	ATTR
Password	String	*YES / *NO *YES means that an FTAC password has been defined	ATTR
AdmPriv	String	*NO	ATTR

[1] Relevant only if UserId is not *STD, *NO is always output in the case of the default admission set. In the normal output, *YES corresponds to an asterisk (*) after the value

8.1.4 FTSHWLOG

The following table indicates the CSV output format of a log record if the INF=*LOGGING-FILES has not been specified. If von INF=*LOGGING-FILES is specified then the output has a different format, see page 389.

The values that are indicated by an “x” in the **Std** column are also output if INF=*STD.

The **Parameter** column contains the name of the output parameter during long output, see page 271ff.

Column	Type	Values and Meaning	Parameter	Std
LogId	Number	Number of the log record (up to twelve digits)	LOGGING-ID	x
ReasonCode	String	Reason code enclosed in double quotes to prevent interpretation as a number. FTAC Reason Codes are output as hexadecimal strings	RC	x
LogTime	Date	Time at which the log record was written	TIME	x
InitUserId	String	Initiator of the request enclosed in double quotes / *REM	INITIATOR	x
InitTsn	String	TSN des Auftraggebers / *NONE	INITSN	x
PartnerName	String	Partner name enclosed in double quotes (name or address)	PARTNER	x
TransDir	String	*TO / *FROM / *NSPEC Transfer direction	TRANS	x
RecType	String	*FT / *FTAC / *ADM Type of log record	REC-TYPE	x
Func	String	*TRANS-FILE / *READ-FILE-ATTR / *DEL-FILE / *CRE-FILE / *MOD-FILE-ATTR / *READ-DIR / *MOVE-FILE / *CRE-FILE-DIR / *DEL-FILE-DIR / *LOGIN / *MOD-FILE-DIR / *REM-ADMIN FT function	FUNCTION	x
UserAdmisId	String	User ID to which the requests in the local system relate, enclosed in double quotes	USER-ADM	x
FileName	String	Local file name enclosed in double quotes	FILENAME	x
Priv	String	*YES / *NO / *NONE Profile is privileged / not privileged / not relevant because no profile was used or no FTAC log record is present	PRIV	
ProfName	String	Name of the FTAC profile enclosed in double quotes / *NONE	PROFILE	
ResultProcess	String	*STARTED / *NOT-STARTED / *NONE Status of follow-up processing	PCMD	

Column	Type	Values and Meaning	Parameter	Std
StartTime	Date	Start time of transfer	STARTTIME	
TransId	Number	Number of transfer request	TRANS-ID	
Write	String	*REPL / *EXT / *NEW / *NONE Write rules	WRITE	
StoreTime	Date	Acceptance time of request – If initiated in the local system: time the request was issued – If initiated in the remote system: time of entry in the request queueh	 REQUESTED STORETIME	
ByteNum	Number	Number of bytes transferred	TRANSFER	
DiagInf	String	Diagnostic information / *NONE	---	
ErrInfo	String	Additional information on the error message, enclosed in double quotes / *NONE	ERRINFO	
Protection	String	*SAME / *STD Protection attributes are transferred / not transferred	PROTECTION ---	
ChangeDate	String	*SAME / *STD Take over modification date of send file for receive file / do not take over modification date	CHG-DATE	
SecEncr	String	*YES / *NO Encryption of request description activated / deactivated	SEC-OPTS	
SecDichk	String	*YES / *NO Data integrity check of request description activated / deactivated	SEC-OPTS	
SecDencr	String	*YES / *NO Encryption of transferred file content activated / deactivated	SEC-OPTS	
SecDdichk	String	*YES / *NO Data integrity check of transferred file content activated / deactivated	SEC-OPTS	
SecLauth	String	*YES / *NO Authentication of the local system in the remote system activated / deactivated	SEC-OPTS	
SecRauth	String	*YES / *NO Authentication of the remote system in the local system activated / deactivated	SEC-OPTS	

Column	Type	Values and Meaning	Parameter	Std
RsaKeyLen	Number	768 / 1024 / 2048 / empty Length of the RSA key used for the encryptio in bit or empty if SecEncr does not have the value *YES	SEC-OPTS	
SymEncrAlg	String	*DES / *AES-128 / *AES-256 / empty The encryption algorithm used or empty if SecEncr does not have the value *YES	SEC-OPTS	
CcsName	String	Name of the character set enclosed in double quotes / empty	CCS-NAME	
AdminId	String	empty	ADMIN-ID	
Routing	String	Routing information enclosed in double quotes / empty	ROUTING	
AdmCmd	String	Administration kommand enclosed in double quotes / empty	ADM-CMD	
As3Type	String	empty (internal function)	---	
As3MsgTid	String	empty (internal function)	---	
As3RcpStat	String	empty (internal function)	---	
AuthLev	Number	1 / 2 / empty Authentication level	SEC-OPTS	
GlobReqId	Number	Global request identification (requests issued remotely) / empty (requests issued locally)	GLOB-ID	

CSV output on INF=*LOGGING-FILES

If the optionINF=*LOGGING-FILES is specified then only the following columns are output:

Column	Type	Values and Meaning	Parameter
TimeStamp	Date	Creation time of the log file	---
LoggingFileName	String	Fully qualified name of the log file	(file name)

8.1.5 FTSHWMON

The following table shows the CSV output format for the monitoring values for openFT operation if all the monitoring values are output (NAME=*ALL,INF=*VALUES(..)).

If DATA=*RAW is specified, the duration values are not output (*Duxxx*, see footnote).

The default values are marked with "x" in the **Std** column. These are output if INF=*STD is specified.

For a detailed description of the monitoring values, refer to thesection “Description of the monitoring values” on page 279.

The individual monitoring values (ThNetbTtl ... StTrcr) have the same names in all the output formats (normal output, long output and CSV output).

Column	Type	Values prepared	Values not prepared	Meaning	Std
CurrTime	Date	Time	Time	Current timet	x
MonOn	Date	Time	Time	Start time of measurement date recording or last change of configuration (a modification of PartnerSel/ReqSel has the same effect as a new start)	x
PartnerSel	String6	*ALL / *NONE / OPENFT / FTP		Partner type selected	x
ReqSel	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE		Request type selected	x
Data	String	FORM	RAW	Output format (perpared / not prepared)	x
ThNetbTtl	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes	x
ThNetbSnd	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, send requests	x
ThNetbRcv	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, receive requests	x
ThNetbTxt	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, text files	
ThNetbBin	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, binary files	
ThDiskTtl	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes	x
ThDiskSnd	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, send requests	x

Column	Type	Values prepared	Values not prepared	Meaning	Std
ThDiskRcv	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, receive requests	x
ThDiskTxt	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, text files	
ThDiskBin	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, binary files	
ThRqto	Number	Number per second	Number, accumulated	openFT requests received	x
ThRqft	Number	Number per second	Number, accumulated	File transfer requests received	
ThRqfm	Number	Number per second	Number, accumulated	file management requests received	
ThSuct	Number	Number per second	Number, accumulated	Successfully completed openFT requests	x
ThAbrt	Number	Number per second	Number, accumulated	Aborted openFT requests	x
ThIntr	Number	Number per second	Number, accumulated	Interrupted openFT requests	x
ThUsrf	Number	Number per second	Number, accumulated	Requests from non-authorized users	x
ThFoll	Number	Number per second	Number, accumulated	Follow-up processing operations started	
ThCosu	Number	Number per second	Number, accumulated	Connections established	
ThCofl	Number	Number per second	Number, accumulated	Failed connection attempts	x
ThCobr	Number	Number per second	Number, accumulated	Disconnections as a result of connection errors	x
DuRqtlOut[1]	Number	Milliseconds	---	Maximum request duration Outbound	
DuRqtlInb[1]	Number	Milliseconds	---	Maximum request duration Inbound	
DuRqftOut[1]	Number	Milliseconds	---	Maximum request duration Outbound transfer	
DuRqftInb[1]	Number	Milliseconds	---	Maximum request duration Intbound transfer	
DuRqfmOut[1]	Number	Milliseconds	---	Maximum request duration Outbound file management	

Column	Type	Values prepared	Values not prepared	Meaning	Std
DuRqfmInb[1]	Number	Milliseconds	---	Maximum request duration Inbound file management	
DuRqesOut[1]	Number	Milliseconds	---	Maximum outbound request waiting time	
DuDnscOut[1]	Number	Milliseconds	---	Maximum time an outbound openFT request was waiting for partner checking	
DuDnscInb[1]	Number	Milliseconds	---	Maximum time an inbound openFT request was waiting for partner checking	
DuConnOut[1]	Number	Milliseconds	---	Maximum duration tim of establishment of a connection for an outbound openFT request	
DuOpenOut[1]	Number	Milliseconds	---	Maximum file open time (outbound)	
DuOpenInb[1]	Number	Milliseconds	---	Maximum file open time (inbound)	
DuClosOut[1]	Number	Milliseconds	---	Maximum file close time (outbound)	
DuClosInb[1]	Number	Milliseconds	---	Maximum file close time (inbound)	
DuUsrcOut[1]	Number	Milliseconds	---	Maximum user check time (outbound)	
DuUsrcInb[1]	Number	Milliseconds	---	Maximum user check time (ínbound)	
StRqas	Number (100)[2]	Average value	Current number	Number of synchronous requests in the ACTIVE state	x
StRqaa	Number (100)[2]	Average value	Current number	Number of asynchronous requests in the ACTIVE state	x
StRqwt	Number (100)[2]	Average value	Current number	Number of requests in the WAIT state	x
StRqhd	Number (100)[2]	Average value	Current number	Number of requests in the HOLD state	x
StRqsp	Number (100)[2]	Average value	Current number	Number of requests in the SUSPEND state	x
StRqlk	Number (100)[2]	Average value	Current number	Number of requests in the LOCKED state	x
StRqfi	Number (100)[2]	Average value	Current number	Number of requests in the FINISHED state	

Column	Type	Values prepared	Values not prepared	Meaning	Std
StCLim	Number	Value currently set		Maximum number of connections established for asynchronous requests.	x
StCAct	Percent	Share of StCLim in %	Current number	Number of occupied connections for asynchronous requests	x
StRqLim	Number	Value currently set		Maximum number of asynchronous requests in request management	x
StRqAct	Percent	Share of StRqLim in %	Current number	Entries occupied in request management	x
StOftr	BOOL	1 / 0		openFT Protocol activated / deactivated	x
StFtmr	BOOL	1 / 0		FTAM Protocol activated / deactivated	x
StFtpr	BOOL	1 / 0		FTP Protocol activated / deactivated	x
StTrcr	BOOL	1 / 0		Trace activated / deactivated	

[1] is not output with DATA=*RAW

[2] number is the measured value multiplied by 100 (e.g. output 225 corresponds to value 2.25)

Examples

```
FTSHWMON NAME=*ALL,OUTPUT=*STDOUT(*CSV)

CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThNetbSnd;ThNetbRcv;ThNetbTxt
;ThNetbBin;ThDiskTtl;ThDiskSnd;ThDiskRcv;ThDiskTxt;ThDiskBin;ThRqto;ThRqft;Th
Rqfm;ThSuct;ThAbrt;ThIntr;ThUsrf;ThFoll;ThCosu;ThCofl;ThCobr;DuRqtlOut;DuRqtl
Inb;DuRqftOut;DuRqftInb;DuRqfmOut;DuRqfmInb;DuRqesOut;DuDnscOut;DuDnscInb;DuC
onnOut;DuOpenOut;DuOpenInb;DuClosOut;DuClosInb;DuUsrcOut;DuUsrcInb;StRqas;StR
qaa;StRqwt;StRqhd;StRqsp;StRqlk;StRqfi;StCLim;StCAct;StRqLim;StRqAct;StOftr;S
tFtmr;StFtpr;StTrcr

2012-07-13 10:44:24;2012-07-13 10:35:46;*ALL;*ALL;FORM;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;5129;0;5129;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1
6;0;2000;0;1;0;1;0
```

8.1.6 FTSHWOPT

The following table indicates the CSV output format of the operating parameters

The **Parameter** column contains the name of the output parameter during normal output, see page 289ff. Some parameters have fixed values because they are supported only for reasons of compatibility or have been replaced by other parameters.

Column	Type	Values and Meaning	Parameter
PartnerLim	Number	0	---
ReqLim	Number	Maximum number of requests	RQ-LIM
TaskLim	Number	Maximum number of processes	PROC-LIM
ConnLim	Number	Maximum number of connections	CONN-LIM
ReqWaitLev	Number	1	---
TransportUnitSize	Number	Maximum length of a transport unit	TU-SIZE
PartnerCheck	String	*STD / *TRANSP-ADDR Partner check	PTN-CHK
SecLev	Number	0... 100 / *B-P-ATTR Default value for the security level of partners	SEC-LEV
TraceOpenft	String	*STD / *OFF Trace function for openFT partner activated / deactivated	FUNCT, line TRACE PARTNER-SELECTION
TraceOut	String	*FILE / empty Trace function activated / deactivated	FUNCT, line TRACE SWITCH---
TraceSession	String	*OFF	---
TraceFtam	String	*STD / *OFF Trace function for FTAM partner activated / deactivated	FUNCT, line TRACE PARTNER-SELECTION
LogTransFile	String	*ON / *OFF FT logging activated / deactivated	FT-LOG
MaxInboundReq	Number	Maximum number of requests	(same as RQ-LIM)
MaxReqLifetime	String	Maximum lifetime of requests in the request queue / *UNLIMITED	MAX-RQ-LIFE
SnmpTrapsSubsystemState	String	*ON / *OFF SNMP traps on subsystem status change activated / deactivated	TRAP, line SNMP SS-STATE
SnmpTrapsFtState	String	*ON / *OFF SNMP traps on asynchronous server status change activated / deactivated	TRAP, line SNMP FT-STATE

Column	Type	Values and Meaning	Parameter
SnmpTrapsPartnerState	String	*ON / *OFF SNMP traps on partner status change activated / deactivated	TRAP, line SNMP PART-STATE
SnmpTrapsPartnerUnreach	String	*ON / *OFF SNMP traps on unreachable partner systems activated / deactivated	TRAP, line SNMP PART-UNREA
SnmpTrapsReqQueueState	String	*ON / *OFF SNMP traps on request management status change activated / deactivated	TRAP, line SNMP RQ-STATE
SnmpTrapsTransSucc	String	*ON / *OFF SNMP traps on successfully terminated requests activated / deactivated	TRAP, line SNMP TRANS-SUCC
SnmpTrapsTransFail	String	*ON / *OFF SNMP traps on failed requests activated / deactivated	TRAP, line SNMP TRANS-FAIL
ConsoleTraps	String	*ON / *OFF Console traps (for at least one criterion) activated / deactivated.	TRAP, line CONS
TeleService	String	empty	
HostName	String	Host name of the local computer / *NONE	HOST-NAME
Identification	String	Instance identification enclosed in double quotes	IDENTIFICATION
UseTns	String	*NO	---
ConsTrapsSubsystemState	String	*ON / *OFF Console traps on subsystem status change activated / deactivated	TRAP, line CONS SS-STATE
ConsTrapsFtState	String	*ON / *OFF Console traps on asynchronous server status change activated / deactivated	TRAP, line CONS FT-STATE
ConsTrapsPartnerState	String	*ON / *OFF Console traps on partner status change activated / deactivated	TRAP, line CONS PART-STATE
ConsTrapsPartnerUnreach	String	*ON / *OFF Console traps on unreachable partner systems activated / deactivated	TRAP, line CONS PART-UNREA
ConsTrapsReqQueueState	String	*ON / *OFF Console traps on request management status change activated / deactivated	TRAP, line CONS RQ-STATE

Column	Type	Values and Meaning	Parameter
ConsTrapsTransSucc	String	*ON / *OFF Console traps on successfully terminated requests activated / deactivated	TRAP, line CONS TRANS-SUCC
ConsTrapsTransFail	String	*ON / *OFF Console traps on failed requests activated / deactivated	TRAP, line CONS TRANS-FAIL
FtLog	String	*ALL / *FAIL / *NONE Scope of FT logging	FT-LOG
FtacLog	String	*ALL / *FAIL / *NONE Scope of FTAC logging	FTAC-LOG
Trace	String	*ON / *OFF Trace function activated / deactivated	FUNCT, line TRACE SWITCH
TraceSelp	String	*ALL / OPENFT / FTP / ADM / empty [1] Trace selection based on partner type	FUNCT, line TRACE PARTNER-SELECTION
TraceSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Trace selection based on request type	FUNCT, line TRACE REQUEST-SELECTION
TraceOpt	String	*NO-BULK-DATA / *NONE Minimum trace / no trace options	FUNCT, line TRACE OPTIONS
KeyLen	Number	768 / 1024 / 2048 RSA key length in bit	KEY-LEN
CcsName	String	empty	---
AppEntTitle	String	*YES Not relevant on z/OS	---
StatName	String	$FJAM	LOCAL-SYSTEM-NAME
SysName	String	Name of the local system	LOCAL-SYSTEM-NAME
FtStarted	String	*YES / *NO openFT started / not started	STARTED
openftAppl	String	*STD / port number Port number of the local openFT server	OPENFT-APPL
ftamAppl	String	*NONE	FTAM-APPL
FtpPort	Number	Port number Port number of the local FTP server	FTP-PORT
ftpDPort	Number	Value / empty (internal function)	---
ftstdPort	String	*STD / port number Default port for dynamic partners	---

Column	Type	Values and Meaning	Parameter
DynPartner	String	*ON / *OFF Dynamic partner entries activated / deactivated	DYN-PART
ConTimeout	Number	Value (internal function)	---
ChkpTime	Number	Value (internal function)	---
Monitoring	String	*ON / *OFF Monitoring data activated / deactivated	FUNCT, line MONITOR SWITCH
MonSelp	String	*ALL / OPENFT / FTP / empty [1] Selection based on type of partner system	FUNCT, line MONITOR PARTNER-SELECTION
MonSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Selection based on type of request	FUNCT, line MONITOR REQUEST-SELECTION
AdmTrapServer	String	Name of the ADM-TRAP server / *NONE	ADM-TRAP-SERVER
AdmTrapsFtState	String	*ON / *OFF ADM traps on asynchronous server status change activated / deactivated	TRAP, line ADM FT-STATE
AdmTrapsPartnerState	String	*ON / *OFF ADM traps on partner status change activated / deactivated	TRAP, line ADM PART-STATE
AdmTrapsPartnerUnreach	String	*ON / *OFF ADM traps on unreachable partner systems activated / deactivated	TRAP, line ADM PART-UNREA
AdmTrapsReqQueueState	String	*ON / *OFF ADM traps on request management status change activated / deactivated	TRAP, line ADM RQ-STATE
AdmTrapsTransSucc	String	*ON / *OFF ADM traps on successfully terminated requests activated / deactivated	TRAP, line ADM TRANS-SUCC
AdmTrapsTransFail	String	*ON / *OFF ADM traps on failed requests activated / deactivated	TRAP, line ADM TRANS-FAIL
AdminConnLim	String	Maximum number of administration connections	ADM-CLIM
AdmPort	String	Port number / *NONE Port number for remote administration	ADM-PORT
OpenftApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the openFT server	OPENFT-APPL, 2nd line
FtamApplState	String	*NAVAIL Status of the FTAM server	FTAM-APPL, 2nd line

Column	Type	Values and Meaning	Parameter
FtpState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the FTP server	FTP-PORT, 2nd line
AdmState	String	*ACTIVE / *INACT / *DISABLED Status for inbound remote administration	ADM-PORT, 2nd line
AdminLog	String	*ALL / *FAIL / *MODIFY / *NONE Scope of ADM logging	ADM-LOG
CentralAdminServer	String	*NO	---
ActiveAppl	String	*ALL / *NONE / OPENFT / FTP / ADM[1] active servers	see 2nd line of OPENFT-APPL, FTAM-APPL, FTP-PORT, ADM-PORT
UseCmx	String	*NO	---
TraceOptLowerLayers	String	*OFF	---
EncMandIn	String	*YES / *NO Inbound encryption activated / deactivated	ENC-MAND (IN)
EncMandOut	String	*YES / *NO Outbound encryption activated / deactivated	ENC-MAND (OUT)
DelLog	String	*ON / *OFF Automatic deletion of log records activated / deactivated	DEL-LOG
DelLogRetpd	Number	Minimum age, in days, of the log records to be deleted. 0 means current day.	RETPD
DelLogRepeat	String	*MONTHLY / *WEEKLY / *DAILY Repeat interval for deletion of log records.	DEL-LOG ON
DelLogDay	Number	1..31 / 1..7 / 0 Day on which deletion is to be repeated. In the case of DelLogRepeat = *MONTHLY then this is the day of the month, if DelLogRepeat = *WEEKLY then it is the day of the week (1 = Monday), if DelLogRepeat = *DAILY then 0 is output	DEL-LOG ON
DelLogTime	Time	Time of deletion	DEL-LOG AT

[1] Combinations of multiple values are also possible (not with *ALL or *NONE)

8.1.7 FTSHWPRF

The following table indicates the CSV output format of an admission profile.

The values that are marked by an “x” in the **Std** column are also output if INF=*ONLY-NAMES is specified.

The **Parameter** column contains the name of the output parameter during long output, see also page 297f.

Column	Type	Values and Meaning	Parameter	Std
ProfName	String	Name of the profile enclosed in double quotes	(Profile name)	x
Priv	String	*YES / *NO Profile is privileged / not privileged	PRIVILEGED	x
TransAdm	String	*SECRET / *NSPEC Transfer admission has been assigned / not assigned	TRANS-ADM NOT-SPECIFIED	x
Duplicated	String	*YES / *NO *YES means: profile is locked due to attempt to assign the transfer admission twice	TRANS-ADM DUPLICATED	x
LockedByImport	String	*YES / *NO *YES means: profile is locked because it was imported	TRANS-ADM LOCKED (by_import)	x
LockedByAdm	String	*YES / *NO *YES means: profile locked by FTAC administrator	TRANS-ADM LOCKED (by_adm)	x
LockedByUser	String	*YES / *NO *YES means: profile locked by user	TRANS-ADM LOCKED (by_user)	x
Expired	String	*YES / *NO *YES means: profile locked because period expired	TRANS-ADM EXPIRED	x
ExpDate	String	Expiration date in short format yyyy-mm-dd / *NRES (no expiration date)	EXP-DATE	
Usage	String	*PUBLIC / *PRIVATE / *NSPEC Usage	USAGE	
IgnObs	String	*YES / *NO Ignore / do not ignore predefined value for Outbound Send	IGN-MAX-LEVELS OBS	
IgnObr	String	*YES / *NO Ignore / do not ignore predefined value for Outbound Receive	IGN-MAX-LEVELS OBR	

Column	Type	Values and Meaning	Parameter	Std
IgnIbs	String	*YES / *NO Ignore / do not ignore predefined value for Inbound Send	IGN-MAX-LEVELS IBS	
IgnIbr	String	*YES / *NO Ignore / do not ignore predefined value for Intbound Receive	IGN-MAX-LEVELS IBR	
IgnIbp	String	*YES / *NO Ignore / do not ignore predefined value for Inbound Processing	IGN-MAX-LEVELS IBP	
IgnIbf	String	*YES / *NO Ignore / do not ignore predefined value for Inbound File Management	IGN-MAX-LEVELS IBF	
Initiator	String	*LOC / *REM / *NRES Initiator: only local / only remote / unrestricted	INITIATOR	
TransDir	String	*FROM / *TO / *NRES Permitted transfer direction: from partner / to partner / unrestricted	TRANS-DIR	
MaxPartLev	Number	0... 100 / *NRES Maximum security level / security level unrestrictedt	MAX-PART-LEV	
Partners	String	One or more FT partners, delimited by commas and enclosed in double quotes / *NRES (no restriction)	PARTNER	
FileName	String	File name or file name prefix enclosed in double quotes / *NRES Restricts access to this file or files with this prefix. *NRES means there is no restriction	FILE-NAME	
Library	String	Library name enclosed in double quotes / *YES / *NO / *NRES Restricts access to this library, *NRES means there is no restriction	LIBRARY	
FileNamePrefix	String	*YES / *NO The file name in FileName is a prefix / is not a prefix	FILE-NAME = (PREFIX=..)	
ElemName	String	Name of the library element enclosed in double quotes / *NONE / *NRES	ELEMENT	
ElemPrefix	String	*YES / *NO The element name in ElemName is a prefix / is not a prefix	ELEMENT	

Column	Type	Values and Meaning	Parameter	Std
ElemVersion	String	Version of the library element enclosed in double quotes / *STD / *NONE / *NRES	ELEMENT	
ElemType	String	Type of the library element enclosed in double quotes / *NONE / *NRES	TYPE	
FilePass	String	*YES / *NRES / *NONE File password	---	
Write	String	*NEW / *EXT / *REPL / *NRES Write rules	WRITE	
UserAdmId	String	User ID enclosed in double quotes	USER-ADM (user-id,...)	x
UserAdmAcc	String	Account number enclosed in double quotes / *FIRST/ *NSPEC / *NRES / *NONE	USER-ADM (..,account,...)	
UserAdmPass	String	*OWN / *YES / *NSPEC / *NONE Password is taken over / was specified / was not specified / is not required	USER-ADM (...,...,password)	
ProcAdmId	String	User ID used for follow-up processing, enclosed in double quotes / *SAME / *NRES	PROC-ADM (user-id,...)	
ProcAdmAcc	String	Account number used for follow-up processing, enclosed in double quotes / *SAME / *NRES / *NONE	PROC-ADM (..,account,...)	
ProcAdmPass	String	*NONE / *YES / *SAME / *NRES Password is taken over / was specified / was not specified /is not required	USER-ADM (...,...,password)	
SuccProc	String	Follow-up processing on success, enclosed in double quotes / *NONE / *NRES / *EXPANSION	SUCC-PROC	
SuccPrefix	String	Folow-up processing prefix on success, enclosed in double quotes / *NONE	SUCC-PREFIX	
SuccSuffix	String	Follow-up processing suffix on success, enclosed in double quotes / *NONE	SUCC-SUFFIX	
FailProc	String	Follow-up processing on error, enclosed in double quotes / *NONE / *NRES / *EXPANSION	FAIL-PROC	
FailPrefix	String	Follow-up processing prefix on error, enclosed in double quotes / *NONE	FAIL-PREFIX	
FailSuffix	String	Follow-up processing suffix on error, enclosed in double quotes / *NONE	FAIL-SUFFIX	

Column	Type	Values and Meaning	Parameter	Std
TransFile	String	*ALLOWED / *NOT-ALLOWED Transfer and delete files permitted / not permitted	FT-FUNCTION = (TRANSFER-FILE)	
ModFileAttr	String	*ALLOWED / *NOT-ALLOWED Modify file attributes permitted / not permitted	FT-FUNCTION = (MODIFY-FILE-ATTRIBUTES)	
ReadDir	String	*ALLOWED / *NOT-ALLOWED View directories permitted / not permitted	FT-FUNCTION = (READ-DIRECTORY)	
FileProc	String	*ALLOWED / *NOT-ALLOWED Preprocessing/postprocessing permitted / not permitted	FT-FUNCTION = (FILE-PROCESSING)	
AccAdm	String	*NOT-ALLOWED	---	
RemAdm	String	*ALLOWED / *NOT-ALLOWED Remote administration via remote administration server permitted / not permitted	FT-FUNCTION = (REMOTE-ADMINISTRATION)	
Text	String	Text enclosed in double quotes / *NONE	TEXT	
DataEnc	String	*YES / *NO / *NRES Data encryption is mandatory / prohibited / neither mandatory nor prohibited	DATA-ENC	
ModDate	Date	Time of last modification	LAST-MODIF	
AdmTrapLog	String	*ALLOWED / *NOT-ALLOWED Reception of ADM traps permitted / not permitted	FT-FUNCTION = (ADM-TRAP-LOG)	

8.1.8 FTSHWPTN

The following table indicates the CSV output format of a partner in the partner list.

The **Parameter** column contains the name of the output parameter during long output, see page 301.

Column	Type	Values and Meaning	Parameter
PartnerName	String	Partner name enclosed in double quotes	NAME
Sta	String	*ACT / *DEACT / *NOCON / *LUNK / *RUNK / *ADEAC / *AINAC / *LAUTH / *RAUTH / *NOKEY / *DIERR / *IDREJ Partner status	STATE
SecLev	String	*STD / *B-P-ATTR / 1...100 Global security level / attribute-specific security level / fixed security level	SECLEV
Trace	String	*FTOPT / *STD / *ON / *OFF Trace setting	TRACE
Loc	Number	Number of locally issued file transfer requests to this partner	LOC
Rem	Number	Number of file transfer requests issued by this partner	REM
Processor	String	Processor name enclosed in double quotes / empty	ADDRESS
Entity	String	Entity name enclosed in double quotes / empty	ADDRESS
NetworkAddr	String	Partner address (network address without port number/selectors) enclosed in double quotes	ADDRESS
Port	Number	Port number	ADDRESS (port number)
PartnerCheck	String	*FTOPT / *STD / *TRANSP-ADDR / *AUTH / *AUTHM / *NOKEY Sender verification	P-CHK
TransportSel	String	Transport selector enclosed in double quotes / empty	ADDRESS (transport selector)
LastAccessDate	Date	Time of last access in short format yyyy-mm-dd	---
SessionSel	String	Session selector enclosed in double quotes / empty	ADDRESS (session selector)
PresentationSel	String	Presentation selector enclosed in double quotes / empty	ADDRESS (presentation selector)
Identification	String	Identification enclosed in double quotes / empty	IDENTIFICATION

Column	Type	Values and Meaning	Parameter
SessRout	String	Routing information enclosed in double quotes / *ID / empty *ID means routing information same as identification	ROUTING
PartnerAddr	String	Partner address (including port number und selectors) enclosed in double quotes	ADDRESS
Check	String	*FTOPT / *STD / *TRANSP-ADDR Partner check	P-CHK
AuthMand	String	*YES / *NO Authentication is mandatory / not mandatory	P-CHK
Priority	String	*LOW / *NORM / *HIGH Priority	PRI
AS3	String	*NO (internal function)	---
AuthLev	Number	1 / 2 / empty Authentication level	P-CHK
InboundSta	String	*ACT / *DEACT Inbound function activated / deactivated	INBND
RequProc	String	*STD / *SERIAL The processing mode for asynchronous outbound requests is parallel / is serial	REQU-P

8.1.9 FTSHWRGE

The following table indicates the CSV output format of partners.

The **Parameter** column contains the name of the output parameter during normal output, see page 307.

Column	Type	Values and Meaning	Parameter
SecLev	Number	Security level	SECLEV
PartnerName	String	Partner name	PARTNER-NAME

8.1.10 NSTATUS

The following table indicates the CSV output format of a request.

Short output is also possible with NSTATUS, see page 410.

The **Parameter** column contains the name of the output parameter during long output, see page 360.

Column	Type	Values and Meaning	Parameter
TransId	Number	Request ID	TRANSFER-ID
Initiator	String	*LOC / *REM Initiator is local / remote	INITIATOR
State	String	*LOCK / *WAIT / *HOLD / *FIN / *ACT / *CANC / *SUSP Request status	STATE
PartnerName	String	Name or address of the partner enclosed in double quotes	PARTNER
PartnerState	String	*ACT / *INACT / *NOCON / *INSTERR Partner status	PARTNER-STATE
TransDir	String	*TO / *FROM Transfer direction	TRANS
ByteNum	Number	Number of bytes transferred / empty	BYTECNT
LocFileName	String	File name or library name in the local system enclosed in double quotes	LOC: FILE or LIBRARY
LocElemName	String	Name of the library element in the local system enclosed in double quotes / *NSPEC	LOC: ELEMENT
LocElemType	String	Type of the library element in the local system enclosed in double quotes / *NSPEC / *NONE	LOC: TYPE
LocElemVersion	String	Version of the library element in the local system enclosed in double quotes / *NSPEC / *NONE	LOC: VERSION
Prio	String	*NORM / *LOW Priority of the request	PRIO
Compress	String	*NONE / *BYTE / *ZIP Compressed transfer	COMPRESS
DataEnc	String	*YES / *NO User data is transferred encrypted / unencrypted	ENCRYPT

Column	Type	Values and Meaning	Parameter
DiCheck	String	*YES / *NO Data integrity is checked / is not checked	DICHECK
Write	String	*REPL / *EXT / *NEW Write rules	WRITE
StartTime	String	Time at which the request is started (format yy-mm-dd hh:mm:ss) / *SOON (request is started as soon as possible)	START
CancelTime	String	Time at which the request is deleted from the request queue (format yy-mm-dd hh:mm:ss) / *NO (no delete time)	CANCEL
Owner	String	Local user ID enclosed in double quotes	OWNER
DataType	String	*CHAR / *BIN / *USER File type	DATA
Transp	String	*YES / *NO Transfer transparent / not transparent	TRANSP
LocTransAdmId	String	User ID for accessing the local system, enclosed in double quotes / *NONE	LOC: TRANS-ADM (USER)
LocTransAdmAcc	String	Account number for the local system / *NONE	LOC: TRANS-ADM=(...account)
LocProfile	String	Name of the admission profile for accessing the local system enclosed in double quotes / *NONE	LOC: TRANS-ADM=(profile)
LocProcAdmId	String	Transfer admission for follow-up processing in the local system enclosed in double quotes / *NONE	LOC: PROC-ADM=(user...)
LocProcAdmAcc	String	Account number for follow-up processing in the local system / *NONE	LOC: PROC-ADM=(...account)
LocSuccProc	String	Local follow-up processing on success, enclosed in double quotes / *NONE / empty	LOC: SUCC-PROC
LocFailProc	String	Local follow-up processing on error, enclosed in double quotes / *NONE / empty	LOC: FAIL-PROC
LocListing	String	*SYSLST / *LISTFILE / *NONE Result list in the local system	LOC: LIST
LocMonjv	String	empty	---

Column	Type	Values and Meaning	Parameter
LocCcsn	String	Name of the character set in the local system enclosed in double quotes / *STD	LOC: CCSN
RemFileName	String	File name in the remote system enclosed in double quotes / *NSPEC / *NONE / empty	REM: FILE or LIBRARY
RemElemName	String	Element name enclosed in double quotes / *NSPEC / *NONE	REM: ELEMENT
RemElemType	String	Element type enclosed in double quotes / *NSPEC / *NONE	REM: TYPE
RemElemVersion	String	Element version enclosed in double quotesn / *STD / *NONE	REM: VERSION
RemTransAdmId	String	User ID in the remote system enclosed in double quotes / *NONE	REM: TRANS-ADM=(user-id,...)
RemTransAdmAcc	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)
RemTransAdmAccount [1]	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)
RemProfile	String	*YES / *NONE *YES means access via FTAC admission profile	REM: TRANS-ADM=REMOTE-PROFILE
RemProcAdmId	String	Transfer admission for follow-up processing in the remote system enclosed in double quotes / *NONE	REM: PROC-ADM=(user-id,...)
RemProcAdmAcc	String	Account number for follow-up processing in the remote system enclosed in double quotes / *NONE	REM: PROC-ADM=(...,account)
RemSuccProc	String	Remote follow-up processing on success, enclosed in double quotes / *NONE / empty	REM: SUCC-PROC
RemFailProc	String	Remote follow-up processing on error, enclosed in double quotes / *NONE / empty	REM: FAIL-PROC
RemCcsn	String	Name of the character set used in the remote system, enclosed in double quotes / *STD	REM: CCSN
FileSize	Number	Size of the file in bytes / empty	FILESIZE
RecSize	Number	Maximum record size in bytes / empty	RECSIZE

Column	Type	Values and Meaning	Parameter
RecFormat	String	*STD / *VARIABLE / *FIX / *UNDEFINED Record format	RECFORM
StoreTime	Date	Time at which the request was entered in the request queue	STORE
ExpEndTime	Date	empty	---
TranspMode	String	*YES / *NO Transfer transparent / not transparent	TRANSP
DataEncrypt	String	*YES / *NO User data transferred encrypted / unencrypted	ENCRYPT
TabExp	String	*AUTO / *YES / *NO Tabulator expansion	TABEXP
Mail	String	*ALL / *FAIL / *NO Result messages	LOC: MAIL
DiagCode	String	empty	---
FileAvail	String	*NSPEC	---
StorageAccount	String	empty	---
AccessRights	String	empty	---
LegalQualif	String	empty	---
PartnerPrio	String	*LOW / *NORM / *HIGH Partner priority	PARTNER-PRIO
TargetFileForm	String	*STD / *BLOCK / *SEQ File format in the target system	TARGFORM
TargetRecForm	String	*STD / *UNDEFINED Record format in the target system	TRECFRM
Protection	String	*STD / *SAME Transfer of protection attributes	PROTECT
GlobReqId	Number	Global request identification For locally issued requests, same as request ID; for globally issued requests, same as the request ID in the initiating system	TRANSFER-ID or GLOB-ID

[1] RemTransAdmAcc and RemTransAdmAccount have the same meaning and the same content. For reasons of compatibility, both parameters are present in the CSV output.

Short output from NSTATUS in CSV format

INF=*SUMMARY outputs a table with two rows indicating the number of requests that have the corresponding status, see also page 356.

Column	Type	Values
Act	Number	Number of requests with the status ACTIVE
Wait	Number	Number of requests with the status WAIT
Lock	Number	Number of requests with the status LOCK
Susp	Number	Number of requests with the status SUSPEND
Hold	Number	Number of requests with the status HOLD
Fin	Number	Number of requests with the status FINISHED
Total	Number	Total number of requests

8.2 FT system messages

The structure of the FT system messages is as follows:

```
FTRnnnn OPENFT: message text
```

or

```
FTCnnnn message text
```

FTRnnnn

is the message code. The message code is 7 characters long.

message text

is the message text. The text appears in uppercase letters. The message text can contain what are known as inserts, e.g. (&00). These parts of the messages are supplied with the current value (e.g. transfer ID) when the message is output.

Additional explanatory information for the message is given under "Meaning"; "Response" tells you what action you should take. These texts are not output with the message.

Messages with the message code **FTRnnnn** (except FTR4nnn) are output for both the FT user and the FT administrator, in the same way as FTAC messages with the message code **FTCnnnn**.

Messages with the message code **FTR4nnn** are only output for the FT administrator. Messages with this message code are only described in the system administrator manual "openFT for z/OS - Installation and Administration".

All message lists are listed with the intention that it should be possible to locate here, any error message that is likely to occur. This lists also contain a range of messages that are only output under very specific (and thus rare) conditions.

On the other hand, this means that you should not always expect to see that message which appears to best characterize the situation.

Asynchronous messages for the user

In addition to the request confirmation or rejection, which is output to the user directly after the command has been entered, the FT system can also send asynchronous messages to the user indicating that file transfer has been terminated.

Further information is given in the section “Messages and return codes automatically issued by openFT for z/OS” on page 108.

Error codes and additional information

Additional error codes and supplementary partner-specific information can be output for some user messages. These provide additional information for troubleshooting.

As a rule, this supplementary information is made up of a return code from the operating system together with a text supplied by the operating system issued in the language set in the operating system. It is appended to the end of the message and is restricted to a length of 64 characters. Longer texts are truncated.

8.2.1 FTR messages

FTR0000 OPENFT: Request (&00) accepted.

Meaning
The command has been stored in the local system's request queue. File transfer will begin once all the resources have been assigned in both the local and remote system.

(&00): transfer ID assigned by the local FT system. You need the transfer ID in case you wish to cancel (NCANCEL) the FT request later.

FTR0005 OPENFT: Request (&00). File '(&01)' transferred.

Meaning
The file transfer request (&00) has been completed successfully. Follow-up processing for both the local and remote system, if requested, has been initiated (provided no error occurred). Local Errors are indicated by a message.

FTR0020 OPENFT:'(&00)' not found.

Meaning
The command has not been executed because the send file is not cataloged or not on a volume of the local system. The command has not been executed because either the send file is not/is no longer, or the receive file is no longer in the catalog or on a volume of the relevant system.

Response
Correct the file name, read in file from tape or restore send file. Repeat the command.

FTR0035 OPENFT: File locked to prevent multiple access.

Meaning
The command has not been executed because either the send file or the receive file is already locked by another process against simultaneous updating.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly.

FTR0041 OPENFT: Request queue full.

Meaning
The command has not been executed because the maximum number of permissible transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR0108 OPENFT: Request (&00). Remote system not accessible.

Meaning
The command could not be accepted because the partner system is currently not available.

Response
Repeat the command later. If the error persists, contact the system or network administrator.

FTR0236 OPENFT: Current instance (&00) no longer found

Meaning
The command was rejected. The instance (&00) could not be found.

FTR0301 OPENFT: Partner '(&00)' entered state NOCON.

Meaning
The partner system (&00) has switched to the state NOCON. This state means that the partner is no longer accessible.

Response
If necessary, check whether the connection to the partner system has been interrupted.

FTR0302 OPENFT: Partner '(&00)' entered state ACTIVE.

Meaning
The partner system (&00) has switched to the state ACTIVE.

Response
For information only.

FTR0303 OPENFT: Partner '(&00)' entered state LUNK.

Meaning
The partner system (&00) has switched to the state LUNK. This state means that the local FT system is not known in the remote FT system.

Response
Ask the remote system's FT administrator to enter the local system in the remote system's network description file/partner list.

FTR0304 OPENFT: Partner '(&00)' entered state RUNK.

Meaning
The partner system (&00) has switched to the state RUNK. The state RUNK means that the remote system is not known in the local transport system.

Response
Make the remote system known on the local system.

FTR0305 OPENFT: Partner '(&00)' entered state INACT.

Meaning
The partner system (&00) has switched to the state INACT. The state INACT means that the FT administrator has locked outbound requests for this partner system.

Response
Remove the lock if necessary.

FTR0306 OPENFT: Partner '(&00)' entered state AINACT.

Meaning
The partner system has switched to the state AINACT. The state AINACT means that the partner system has been automatically deactivated because a certain number of consecutive connection attempts have failed.

Response
Check whether partner system should be accessible and reactivate the partner system.

FTR0307 OPENFT: Partner '(&00)' may be unreachable.

Meaning
A number of consecutive attempts to connect to the partner system (&00) have failed. Further attempts will be made.

Response
For information only.

FTR0308 OPENFT: Partner '(&00)' does not allow more inbound requests.

FTR0309 OPENFT: Partner '(&00)' added.

Meaning
The specified remote system has been entered in the partner list.

FTR0310 OPENFT: Partner '(&00)' removed.

Meaning
The specified remote system has been removed from the partner list.

FTR0311 OPENFT: Partner '(&00)' entered state LAUTH.

Meaning
The partner system (&00) has switched to the state LAUTH. The state LAUTH means that the local system could not authenticate itself at the remote system.

Response
Send the current key file to the administrator of the remote system.

FTR0312 OPENFT: Partner '(&00)' entered state RAUTH.

Meaning
The partner system (&00) has switched to the state RAUTH. The state RAUTH means that the remote system could not authenticate itself at the local system. This may either be due to an out-of-date key in the key file or to may indicate an access attempt by an unauthorized system.

Response
Contact the administrator of the remote system.

FTR0313 OPENFT: Partner '(&00)' entered state DIERR.

Meaning
The partner system (&00) has switched to the state DIERR. File integrity errors have been detected on the transmission path. This may also indicated deliberate manipulation of the transmission data.

FTR0314 OPENFT: Partner '(&00)' entered state NOKEY.

Meaning
The partner system (&00) has switched to the state NOKEY. The state NOKEY means that the partner will not accept a connection without encryption or that no key is present.

Response
Generate a new key pair.

FTR0315 OPENFT: Partner '(&00)' entered state IDREJ.

Meaning
The partner system (&00) has switched to the state IDREJ. The local identification was not accepted by the local identification or by an intermediate entity.
Possible causes:
– both the local identification and the migrated ID %.<processor>.<entity> are entered in the remote system's request file.
– the identification has been rejected by an intermediate entity for security reasons

Response
Ask for your entity's partner entry to be checked.

FTR0320 OPENFT: Abnormal termination initiated.

Meaning
Abnormal termination of FT has been initiated due to an internal error.

Response
Check the cause of the abnormal termination and restart FT.

FTR0330 OPENFT: Request queue 85 percent full.

Meaning
Approximately 85% of the spaces for request storage in the request file are occupied. Issuing a number of additional requests could completely fill the request queue with the result that FT will reject new requests.

Response
If necessary, increase the size of the request queue.

FTR0331 OPENFT: At least 20 percent of request queue unoccupied.

Meaning
At least 20% of the FT request queue is available. This message is only output if a previous FTR0330 message has warned of a possible queue overflow. The threat of a bottleneck has receded.

FTR0340 OPENFT: Transfer '(&00)' successfully completed.

Meaning
The request designated in greater detail by the insert (&00) has been terminated successfully.
(&00): *LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character '*' at the end of the file name.

Response
For information only.

FTR0341 OPENFT: Transfer '(&00)' terminated with error.

Meaning
The request designated in greater detail by the insert (&00) terminated with an error
(&00): MSGNR;*LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character '*' at the end of the file name.

Response
For information only.

FTR0360 OPENFT: openFT control process started

Response
For information only.

FTR0361 OPENFT: openFT control process terminated

Response
For information only.

FTR0500 OPENFT: openFT (&00) started. Protocols: (&01).

Meaning
The openFT file transfer system openFT has been activated for the protocols (&01).

FTR0501 OPENFT: openFT terminated.

Meaning
The file transfer system openFT has been terminated by means of an administration command.

FTR0502 OPENFT: No log records available for the selection criteria.

Meaning
No logging records meet the selected criteria.

Response
Change the selection criteria.

FTR0503 OPENFT: No partner available for the selection criteria.

Meaning
There are no partners that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0504 OPENFT: No requests available for the selection criteria.

Meaning
There are no requests that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0505 OPENFT: Requests carried out; (&00) files were transferred

Meaning
The file transfer requests have been successfully completed. A total of (&00) files have been transferred. If you have specified commands for follow-up processing, follow-up processing is carried out for every file.

FTR0560 OPENFT: Cancel all specified requests? Reply (y=yes; n=no)

Meaning
A CANCEL-TRANSFER command applies to more than one file transfer.
Y: All the transfer requests affected are deleted.
N: The entire deletion request is withdrawn.

FTR0562 OPENFT: (&00):

FTR0600 OPENFT: Shutdown processing delayed. FT tasks pending.

Meaning
openFT could not be terminated.

Response
Check if there are console messages that need to be answered for FT tasks connected to the FT subsystem.

FTR0604 OPENFT: Request (&00). Follow-up processing not started.

Meaning
The follow-up processing of a transfer request was not started because the local processing admission may be incorrect.

Response
Correct the local processing admission and repeat the command.

FTR0605 OPENFT: Tracefile changed

Meaning
There has been a switch to a new trace file.

FTR0606 OPENFT: Trace terminated.

Meaning
The trace status has been switched off.

FTR0607 OPENFT: Trace started: (&00).

Meaning
The trace status for the protocols specified in (&00) has been switched on.

FTR0700 Parameter '(&00)' and '(&01)' must not be specified at the same time

Meaning
The selected parameters could not be specified simultaneously.

Response
Omit one of the two parameters and repeat the command.

FTR0701 OPENFT: Input error

FTR0702 OPENFT: Parameter value '(&00)' too long

Meaning
The specified parameter value (&00) is too long; see the command syntax.

Response
Reduce the length of the parameter value (&00) and repeat the command.

FTR0703 OPENFT: Mandatory parameter missing

Meaning
A mandatory parameter is missing; see the command syntax.

Response
Correct the command and try again.

FTR0704 OPENFT: Mandatory parameter '(&00)' missing

Meaning
The mandatory parameter (&00) was not specified.

Response
Correct the command and try again.

FTR0705 OPENFT: Parameter '(&00)' specified more than once

Meaning
The parameter (&00) was specified more than once.

Response
Correct the command and try again.

FTR0706 OPENFT: Parameter '(&00)' can only be specified together with '(&01)'

Meaning
The parameter (&00) can only be specified together with (&01).

Response
Add the parameter (&01) to the command and repeat the command.

FTR0707 OPENFT: Invalid parameter '(&00)'

Meaning
An invalid parameter (&00) was specified; see the command syntax.

Response
Correct the command and try again.

FTR0708 OPENFT: Value of parameter '(&00)' not within valid range

Meaning
The parameter value (&00) is not within the specified value range; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0709 OPENFT: Too many positional parameters

Meaning
The maximum number of positional parameters was exceeded.

Response
Correct the command and try again.

FTR0710 OPENFT: Invalid parameter value '(&00)'

Meaning
The assigned parameter value (&00) is incorrect; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0750 OPENFT: Command not found

FTR0751 OPENFT: Command name ambiguous with regard to '(&00)'

FTR0752 OPENFT: Closing parenthesis missing for operand '(&00)'

FTR0753 OPENFT: Invalid delimiter '(&00)' after operand '(&00)'

FTR0755 OPENFT: List value of operand '(&00)' is not consistent with data type '(&00)'

FTR0756 OPENFT: Operand value introducing the structure is mandatory for '(&00)'

FTR0757 OPENFT: Value of operand '(&00)' is not consistent with data type '(&00)'

FTR0758 OPENFT: Keyword value of operand '(&00)' is ambiguous with regard to '(&00)'

FTR0759 OPENFT: Too many closing parentheses

FTR0760 OPENFT: The mandatory operand '(&00)' is missing

FTR0762 OPENFT: Operand name '(&00)' ambiguous with regard to '(&00)'

FTR0763 OPENFT: Operand '(&00)' is not known

FTR0764 OPENFT: Operand '(&00)' specified more than once

FTR0765 OPENFT: Too many list elements for operand '(&00)'

FTR0766 OPENFT: Too many positional operands

FTR0767 OPENFT: Too many positional operands for '(&00)'

Meaning
(applies to FTR0750 through FTR0767)
An operand value that introduces a structure can only be omitted if there is only one possible structure specification for the corresponding operand or if this structure specification is the default value for the operand.
The following command, for example, will be rejected with this message:

```
FTMODPRF MYPROF01,PARTNER=((REMSYS1,REMSYS2))
```

Reason: It is not clear which of the following specifications is meant:

```
FTMODPRF MYPROF01,PARTNER=*ADD((REMSYS1,REMSYS2))
```

or

```
FTMODPRF MYPROF01,PARTNER=*REM((REMSYS1,REMSYS2))
```

Response
Repeat the command using the correct syntax.

FTR0780 OPENFT: Internal error: operand buffer overflow

FTR0781 OPENFT: Internal error: structure nesting too deep

FTR0790 OPENFT: Available commands: '(&00)'

FTR0791 OPENFT: Available list-values: '(&00)'

FTR0792 OPENFT: Available operands: '(&00)'

FTR0793 OPENFT: Available values: '(&00)'

FTR0801 OPENFT: Request (&00). Internal error

Meaning
NDMS, FJAM or operating system error that is neither a DMS error nor a transport system error, possibly the transfer ID.
The FT system continues to run after the message has been issued.

FTR0802 OPENFT: Request (&00). Warning: Monitor file contents inconsistent

Meaning
At the end of the file transfer request, the contents of the job variable monitoring the request were found to be inconsistent.
Possible reason: During the transfer, the job variable was accessed externally in a mode other than read mode.
The result of the transfer is not affected and is given in the result list or asynchronous end message.

FTR0803 OPENFT: Request (&00). Follow-up processing could not be started.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands are incorrect.

Response
Define the required PROCESSING ADMISSION or correct it. Repeat the command if necessary.

FTR0804 OPENFT: Request (&00). Request data inconsistent.

FTR0851 OPENFT: Internal error.

FTR0852 OPENFT: Internal error. Current instance '(&00)' incompatible.

Meaning
The system data was not created with the version of the openFT file transfer system currently in use.

Response
Update the instance to the current openFT version using the appropriate command (FJGEN).

FTR0854 OPENFT: Writing of log records no more possible. Process terminated.

Meaning
There is not enough space on the disk/partition on which the logging files are stored.

Response
Increase the disk space (or have it increased).

FTR0855 OPENFT: No space left on device for internal files.

Meaning
There is not enough space on the disk/partition on which the internal files are stored.

Response
Increase the disk space (or have it increased).

FTR0856 OPENFT: Error during ops generation.

FTR0857 OPENFT: Error in key file (&00)

FTR0858 OPENFT: Internal error. Set / release file-locks not possible

Meaning
A problem occurred when setting/resetting the file locks for all open requests in FT-REQUEST-FILE.

Response
Check whether the request file SYSRQF is accessible on the config user ID of the current instance.

FTR0862 OPENFT: Protocol stack (&00) not installed

Meaning
The required transfer protocol is not installed.

Response
Install the transfer protocol.

FTR0863 OPENFT: FTAC subsystem not available

Meaning
Install openFT-AC.

FTR0999 OPENFT: openFT panic (&00). Abnormal termination

FTR1020 OPENFT: openFT already started.

Meaning
openFT can only be started once in each instance.

Response
Terminate openFT if necessary.

FTR1021 OPENFT: Request must be canceled without FORCE option first

Meaning
Before the FORCE option is used, the command must be called without the FORCE option.

Response
Issue the command without the FORCE option first.

FTR1029 OPENFT: Maximum number of key pairs exceeded.

Meaning
The maximum number of key pair sets has been reached.

Response
Before new key pair set can be created, an older key pair set must be deleted.

FTR1030 OPENFT: Warning: last key pair deleted.

Meaning
The last key pair set has been deleted. Without a key pair set, encrypted transfer, authentication and data integrity checking are not possible.

Response
Create a new key pair set.

FTR1031 OPENFT: No key pair available.

Meaning
All transfers are carried out without encryption.

Response
Create a new key pair set, if necessary.

FTR1032 OPENFT: Last key pair must not be deleted

FTR1033 OPENFT: The public key files could not be updated.

Meaning
The contents of the SYSPKF file could not be fully updated.
Possible reasons:
– The SYSPKF file is locked.
– There is not enough disk space to allow the file to be created.

Response
Take the appropriate action depending on the cause of the error:
– Unlock the file.
– Allocate disk space or have your system administrator do it.

Update the key with UPDATE-FT-PUBLIC-KEY.

FTR1034 OPENFT: Command only permissible for FT or FTAC administrator

Meaning
Only the FT or FTAC administrator is permitted to use the command.

Response
Have the command executed by the FT or FTAC administrator.

FTR1035 OPENFT: Command only permissible for FT administrator.

Meaning
Only the FT administrator is permitted to use the command.

Response
Have the command executed by the FT administrator.

FTR1036 OPENFT: User not authorized for other user Ids.

Meaning
The user is not authorized to use a different user ID in the command.

Response
Specify your own ID, or have the command executed by the FT or FTAC administrator.

FTR1037 OPENFT: Key reference unknown.

Meaning
The specified key reference is unknown.

Response
Repeat the command with an existing key reference.

FTR1038 OPENFT: Request '(&00)' is in the termination phase and can no longer be canceled

FTR1039 OPENFT: openFT not active.

Meaning
openFT is not started.

Response
Start openFT, if necessary.

FTR1040 OPENFT: Config user ID unknown or not enough space

Meaning
The CONFIG USERID of the current instance (SYSFJAM) is unknown or the disk space allocated is insufficient to allow creation of the FT-REQUEST-FILE, the file for storing trace data, or the key files.

Response
Either create the CONFIG-USERID or increase its disk space allocation or have your system administrator do it.

FTR1041 OPENFT: Specified file is not a valid trace file

FTR1042 OPENFT: openFT could not be started

FTR1043 OPENFT: Partner with same attribute '(&00)' already exists in partner list.

Meaning
There is already a partner entry with the same attribute '(&00)' in the partner list.

Response
The attribute '(&00)' in partner entries must be unique. Correct the command accordingly and try again.

FTR1044 OPENFT: Maximum number of partners exceeded.

Meaning
The partner list already contains the maximum permissible number of partner entries.

Response
Delete partners that are no longer required.

FTR1045 OPENFT: No partner found in partner list.

Meaning
A partner for the specified selection could not be found in the partner list.

Response
Check if the specified partner name or address was correct. If necessary, repeat the command using the correct name or address.

FTR1046 OPENFT: Modification of partner protocol type not possible

Meaning
The protocol type of the partner entry cannot be changed subsequently.

Response
Delete the partner from the partner list, if necessary, and enter it again with a new protocol type.

FTR1047 OPENFT: Request (&00) not found.

Meaning
The request with the transfer ID (&00) could not be found.

Response
Specify the existing transfer ID and repeat the command.

FTR1048 OPENFT: Active requests could not yet be deleted

Meaning
Active requests for the specified partner were cancelled. After the negotiation of termination with the partner the requests will be automatically deleted.

FTR1049 OPENFT: CCS name (&00) unknown

FTR1057 OPENFT: Inbound requests cannot be modified

FTR1059 OPENFT: Monitoring is not active

Meaning
The command is only supported if monitoring is activated.

Response
Activate monitoring in the operating parameters.

FTR1065 OPENFT: File not found

FTR1066 OPENFT: Not enough space for file

FTR1067 OPENFT: Syntax error in resulting file name

FTR1068 OPENFT: Access to file denied (&00)

FTR1069 OPENFT: Error accessing file (&00)

FTR1076 OPENFT: selected key file not found

FTR1078 OPENFT: Too short time interval since last logging file switch

Meaning
At this moment the logging file cannot be switched, as the time dependant part of the logging file name does not differ from this name part in the actual logging file name.

Response
If necessary, repeat the command after an appropriate waiting time.

FTR1082 OPENFT: User data encryption not supported

Meaning
User data encryption is supported only when openFT-CR is installed.

Response
Install openFT-CR

FTR1083 OPENFT: Structure of key file not supported

Meaning
The key cannot be imported because of the not supported key file structure.

FTR1084 OPENFT: Invalid password

FTR1085 OPENFT: Password missing

FTR1086 OPENFT: Duplicate key pair

Meaning
No import of duplicate keys allowed.

FTR1087 OPENFT: Key expired

Meaning
The expiration date lies in the past.

FTR2014 OPENFT: No file attribute changes requested.

Meaning
No further file attributes besides the file name were specified.

Response
Enter the desired file attributes in addition to the file name.

FTR2015 OPENFT: openFT is not authorized to execute requests for this user

FTR2016 OPENFT: Directory (&00) is not empty

FTR2017 OPENFT: File attributes do not match request parameters (&00)

Meaning
The specified attribute combination is not permissible.

Response
Specify a permissible combination.

FTR2018 OPENFT: Attributes could not be modified (&00).

Meaning
The properties of the file could not be changed as specified in the command.
The following reasons are possible:
For the remote file:
– No access rights to the file.
– The required combination of access rights is not supported by the remote system.

– If the remote system is a BS2000: the file is protected by ACL.

For the local file:
– No access rights to the file.
– The requested transfer attributes are not compatible with the properties of the file (see manual).

FTR2019 OPENFT: (&00)' could not be created (&01).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receiving system's TRANSFER-ADMISSION to the ID of the receive file's owner. Repeat the command.

FTR2021 OPENFT: CCS name unknown.

Meaning
The request could not be completed because the CCS name specified for the local file does not correspond to any of the supported code tables.

FTR2022 OPENFT: Higher-level directory not found

Meaning
In the case of a receive request, the local file could not be created because the specified path does not exist.

Response
Create or correct the path for the receive file and repeat the command.

FTR2023 OPENFT: (&00)' already exists.

Meaning
The command was not executed because an existing receive file cannot be created again with WRITE-MODE=NEW. WRITE-MODE=NEW may also have been set due to a restriction in the access authorization used.

Response
Either delete the receive file and repeat the command, or repeat the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2024 OPENFT: Transfer of file generation groups not supported.

Meaning
The command was not executed because the FT system only transfers single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2025 OPENFT: Error accessing '(&00)'(&02).

Meaning
(&02): Further details, possibly DMS error
The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2026 OPENFT: Resulting file name '(&00)' too long (&01).

Meaning
The relative file name was specified in the transfer request. The absolute file name completed by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2027 OPENFT: No file or directory name specified.

Meaning
The command was not executed because the file name was neither specified explicitly nor by the 'TRANSFER-ADMISSION' used.

Response
Repeat the command, specifying the file ID explicitly or a TRANSFER-ADMISSION that defines the file ID.

FTR2028 OPENFT: Invalid management password.

FTR2029 OPENFT: (&00)' not available (&01).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the local transfer admission (TRANSFER-ADMISSION operand).

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the local transfer admission. Repeat the command.

FTR2030 OPENFT: Home directory not found (&00)

FTR2031 OPENFT: Renaming not possible (&00)

FTR2032 OPENFT: Not enough space for (&00).

Meaning
The command was not (fully) executed because the permissible storage space on the receive system is used up for the user ID specified in TRANSFER-ADMISSION.
The receive file can not be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2033 OPENFT: File owner unknown.

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined in the local system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2034 OPENFT: Invalid file password.

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2036 OPENFT: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2037 OPENFT: '(&00)' is read only.

FTR2038 OPENFT: File structure not supported (&00).

FTR2039 OPENFT: Syntax error in resulting file name '(&00)' (&01).

Meaning
The local file cannot be accessed because, for example, the absolute file name is too long.

Response
Shorten the path or file name. Repeat the command.

FTR2040 OPENFT: Transparent file transfer not supported.

Meaning
The request could not be carried out because the partner system does not support the receipt of files in a transparent format.

FTR2042 OPENFT: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2043 OPENFT: Access to '(&00)' denied (&01).

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only).

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2044 OPENFT: Follow-up processing exceeds length limit.

Meaning
Prefix + suffix (from prof) + local follow-up processing together are too long.

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2045 OPENFT: Processing admission invalid.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands were incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary.

FTR2046 OPENFT: Local transfer admission invalid.

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands were incorrect.

Response
Define the required TRANSFER ADMISSION or correct it.
Repeat the command if necessary.

FTR2047 OPENFT: Request rejected by local FTAC.

Meaning
The command was not executed because the request was rejected by the product openFT-AC due to a lack of authorization.

Response
Use the return code in the logging record to determine and remove the cause.
Repeat the command.

FTR2048 OPENFT: Function not supported for protocol '(&00)'.

Meaning
The desired function is not available for the selected protocol.

Response
Select a different protocol.

FTR2049 OPENFT: Remote follow-up processing not supported

Meaning
Remote follow-up processing is only available for the openFT protocol.

Response
Select a different protocol, or specify follow-up processing by means of an FTAC profile.

FTR2050 OPENFT: Data integrity check not supported.

Meaning
The partner system does not support the data integrity check function.

Response
Repeat the request without a file integrity check.

FTR2051 OPENFT: User data encryption not possible for this request.

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2052 OPENFT: Administration request rejected by remote administration server

Meaning
The command was not executed because the request was rejected by the remote administration server.

Response
Use the return code in the log record on the remote administration server to determine and remove the cause. Repeat the command.

FTR2053 OPENFT: Destination format not supported for transparent transfer

Meaning
The destination file organization parameter is not supported for transparent transfer

Response
Repeat the request without destination file organization parameter.

FTR2054 OPENFT: Invalid command

Meaning
The specified command is not allowed in this context.

Response
Repeat the request with a valid command.

FTR2056 OPENFT: Syntax error in partner name (&00)

Meaning
The syntax of the partner name is wrong.

Response
Correct partner name. Repeat the command.

FTR2058 OPENFT: User data encryption is mandatory

Meaning
The data encryption function is mandatory.

Response
Repeat the request with data encryption.

FTR2070 OPENFT: Request (&00). openFT is no longer authorized to execute requests for this user

FTR2071 OPENFT: Request (&00). User data encryption not installed.

Meaning
The user data encryption function cannot be used unless openFT-CR is installed.

Response
Use openFT-CR.

FTR2072 OPENFT: Request (&00) has been canceled.

Meaning
The FT request was canceled because
– the command NCANCEL was specified, or
– the time specified in NCOPY has been reached.

Follow-up processing has been started for the local system, provided no error occurred. Follow-up processing is started for the remote system once all the resources are allocated. Local errors are indicated by the message FTR0604 at the start of follow-up processing.

FTR2074 OPENFT: Request (&00). '(&01)' could not be created (&02).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2075 OPENFT: Request (&00). Higher-level directory no longer found

FTR2076 OPENFT: Request (&00). I/O error for '(&01)'(&02).

Meaning
The file can no longer be accessed. It may have been deleted during a transfer.

Response
Repeat the request.

FTR2077 OPENFT: Request (&00). File now locked to prevent multiple access.

Meaning
The command was not executed because the send file or the receive file is already locked by another process so that it cannot be simultaneously updated.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2078 OPENFT: Request (&00). '(&01)' no longer available (&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk.

Response
Inform the operator if necessary.
Repeat the command.

FTR2079 OPENFT: Request (&00). '(&01)' no longer found.

Meaning
The local send or receive file can no longer be accessed because, for example, it was deleted during an interruption of the openFT system.

Response
Restore the file.
Repeat the command.

FTR2080 OPENFT: Request (&00). Home directory no longer found (&01)

FTR2081 OPENFT: Request (&00). '(&01)' gets no more space.

Meaning
The command was not executed (any further) executed because
– the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
– the receive file has already reached the maximum number of allocations.
Take the appropriate action depending on the cause of the error:

Response
delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– reorganize the file so that it requires fewer allocations, or
– increase the receive file's primary/secondary allocation.
If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2082 OPENFT: Request (&00). File owner no longer known.

Meaning
The command was not executed because the owner of the send file or receive file is not defined on the relevant system or because the file owner and the user who wants to create a receive file are not the same.

Response
Define the file owner, or correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2083 OPENFT: Request (&00). Pre-/post-processing error(&01).

Meaning
The command executed as part of local pre-/post-processing returned a result other than OK.

Response
Correct and repeat the command.

FTR2084 OPENFT: Request (&00). Exit code (&01) for pre-/post-processing (&02).

Meaning
The command executed as part of local pre-/post-processing returned the exit code (&01).

Response
Correct the command using the exit code (&00) and issue it again.

FTR2085 OPENFT: Request (&00). File password no longer valid.

Meaning
The command was not executed because the password for send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2086 OPENFT: Request (&00). '(&01)' is now read only.

FTR2087 OPENFT: Request (&00). File structure error(&01).

Meaning
The command was executed due to a file structure error.
File structure errors include:
- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If WRITE-MODE=EXTEND-FILE or -e is specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- The send file or receive file in a remote BS2000 system is a member of an old LMS library (not PLAM).

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE or -e is specified, restore the receive file. Repeat the command.

FTR2088 OPENFT: Request (&00). NDMS error (&01).

Meaning
The request was rejected because the partner system currently does not have the resources available to accept requests.

Response
Repeat the request a little later.

FTR2089 OPENFT: Request (&00). Recovery failed (&01).

Meaning
The restart attempts were unsuccessful (for example, a pre-/post-processing command could not be completed before the termination of openFT).

Response
Repeat the command.

FTR2090 OPENFT: Request (&00). Error in file transfer completion.

Meaning
An error occurred during the final phase of the file transfer. If it was a long transfer, the recipient is advised to check if the file has still been transferred correctly. However, error follow-up processing will be started if it was specified.

Response
Repeat the request, if necessary.

FTR2091 OPENFT: Requests only partially completed; (&00) of (&01) files were transferred

Meaning
In the case of a synchronous send request with wildcards, not all files were successfully transferred.

Response
Transfer unsuccessfully transferred files again.

FTR2092 OPENFT: Request (&00). Access to '(&01)' no longer permissible (&02).

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only) or because a directory was specified as either the source or destination of a file transfer.

Response
Correct the transfer direction, write mode, file name or file protection attributes.
Repeat the command.

FTR2094 OPENFT: Request (&00). Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2095 OPENFT: Request (&00). Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2096 OPENFT: Request (&00). File structure not supported (&01).

FTR2097 Request (&00). Resulting file name '(&01)' too long(&02)

Meaning
The relative file name was specified in the transfer request. The absolute file name as extended by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2109 OPENFT: Request (&00). Connection setup rejected by local transport system.

FTR2110 OPENFT: Request (&00). Data integrity check indicates an error.

Meaning
The integrity of the data was violated.

FTR2111 OPENFT: Encryption/data integrity check not possible. Encryption switched off.

Meaning
There is no key pair set or the key length was set to 0. Requests can only be carried out without data encryption or a data integrity check.

Response
Repeat the request without data encryption, create a key or set a key length >0.

FTR2112 OPENFT: Request (&00). Data integrity check not supported by partner.

Meaning
The partner system does not support the data integrity check.

Response
Repeat the request without a data integrity check.

FTR2113 OPENFT: Request (&00). User data encryption not possible for this request.

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2114 OPENFT: Request (&00). Identification of local system rejected by remote system '(&01)'.

Meaning
For security reasons or because of an inconsistency, the partner did not accept the instance identification of the local system (for example, because in a network description file both the instance identification and migration identification %.prozessor.entity occur for different partners).

Response
Ensure that the local identification has been entered correctly on the partner system and has not been assigned to a different partner.

FTR2115 OPENFT: Request (&00). Interrupted by remote system

FTR2116 OPENFT: Local application (&00) not defined

Meaning
The local application is not defined in the transport system, or the tnsxd process will not run in the Unix system.

Response
Make the local application known to the local transport system or start the tnsxd process.

FTR2117 OPENFT: Local application (&00) not available

FTR2118 OPENFT: Request (&00). Authentication of local system failed.

Meaning
The local system could not be authenticated by the partner system.

Response
Give the current public key file to the partner and name it correctly there. Repeat the command.

FTR2119 OPENFT: Request (&00). Local system unknown in remote system.

Meaning
The local system is not known on the partner system (e.g. BS2000/OSD or z/OS).

Response
Make the local system known on the partner system and repeat the command.

FTR2120 OPENFT: Remote system '(&00)' unknown.

Meaning
The partner specified as the remote system cannot be expanded to an address on the local system.

Response
Correct the specification for the partner or add the partner to the partner list and repeat the command.

FTR2121 OPENFT: Request (&00). Authentication of partner failed.

Meaning
The remote system could not be authenticated by the local system.

Response
Get the current public key file from the partner and name it correctly.

FTR2122 OPENFT: Request (&00). FT session rejected or disconnected. Reason (&01)

FTR2123 OPENFT: Request (&00). OSS call error (&01).

Meaning
The command was not executed because the session instance detected a communication error.
(&00): error code.

Response
Take the appropriate action in accordance with the error code.

FTR2124 OPENFT: Request (&00). No free connection

Meaning
No more transfers are possible because the maximum number of simultaneous transfers has been reached.

Response
Check whether the transport system is working (or have it checked).

FTR2125 OPENFT: Request (&00). Connection lost.

Meaning
No data transfer took place because of a line interrupt or a line protocol error.

Response
Repeat the request.

FTR2126 OPENFT: Request (&00). Transport system error. Error code (&01)

Meaning
An error occurred in the transport system during processing of a FTSTART command or a file transfer or file management request.

Response
Take the appropriate action in accordance with the error code. Most often the occurrence of this message indicates that the partner addressed is not known to the transport system.

FTR2127 OPENFT: Request (&00). No data traffic within (&01) seconds

Meaning
No data transfer took place within the period of seconds specified because, for example, the connection is interrupted, the partner is not sending and the local system is waiting for data.

Response
Repeat the request.

FTR2128 OPENFT: OSS version not supported

Meaning
At least OSS version V04.1 required.

FTR2140 OPENFT: Request (&00). Remote system: openFT is not authorized to execute requests for this user.

FTR2141 OPENFT: Request (&00). Remote system: Directory (&01) is not empty

Meaning
The command could not be executed because there are files in the specified directory of the partner system.

Response
Delete all the files in the directory first and repeat the command.

FTR2142 OPENFT: Request (&00). Remote system: File attributes do not match the request parameters (&01)

Meaning
The command could not be executed because the file attributes on the remote system do not agree with the request parameters (e.g. a directory was specified instead of a remote file).

Response
Check the file name on the remote system and correct it. Repeat the command.

FTR2143 OPENFT: Request (&00). Remote system: Attributes could not be modified (&01).

Meaning
The properties of the file could not be modified as desired in the command.
Possible reasons are for the remote file:
– No access rights to the file.
– The combination of access rights required is not supported by the remote system.
– If the remote system is a BS2000: the file is protected by ACL.

FTR2144 OPENFT: Request (&00). Remote system: File/directory (&01) could not be created (&02)

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2145 OPENFT: Request (&00). Remote system: CCS name unknown or not supported.

Meaning
The request could not be completed because the CCS is unknown in the partner system.

FTR2146 OPENFT: Request (&00). Remote system: Higher-level directory not found

Meaning
The command was not executed because the higher-level directory could not be found on the partner system.

Response
Create the directory on the remote system or correct the remote directory name and repeat the command.

FTR2147 OPENFT: Request (&00). Remote system: File/directory '(&01)' already exists.

Meaning
The command was not executed. Possible reasons:
– The command was not executed because an existing receive file cannot be created with 'WRITE-MODE=NEW' or the -n option. WRITE-MODE=NEW or -n may also have been set by a restriction in the access authorization used.
– ftcredir: The specified directory already exists.

Response
Either delete the receive file before repeating the command or reenter the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2148 OPENFT: Request (&00). Remote system: Transfer of file generation groups not supported.

Meaning
The command was not executed because the FT system can only transfer single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2149 OPENFT: Request (&00). Remote system: Access error for '(&01)' (&02).

Meaning
(&02): DMS error, possibly the transfer ID. The FT system continues to run after output of the message.

Response
Take the appropriate action in accordance with the error code.

FTR2150 OPENFT: Request (&00). Remote system: Resulting file name too long (&01).

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected. Possible reasons:
– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Response
Repeat the command using the correct syntax.

FTR2151 OPENFT: Request (&00). Remote system: File locked to prevent multiple access.

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously.

Response
Repeat the command later or unlock the file on the remote system. After a system crash in BS2000 you may need to verify files not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2152 OPENFT: Request (&00). Remote system: No file or directory name specified.

Meaning
The command was not executed because the file ID was neither specified explicitly nor by Repeat the command, specifying the file ID explicitly or using a TRANSFER ADMISSION that defines the file ID.

FTR2153 OPENFT: Request (&00). Remote system: Invalid management password.

FTR2154 OPENFT: Request (&00). Remote system: File/directory '(&01)' not available (&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the remote transfer admission.

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the remote transfer admission. Repeat the command.

FTR2155 OPENFT: Request (&00). Remote system: File/directory '(&01)' not found.

Meaning
The command was not executed because the send file is not or no longer in the catalog or on a volume of the remote system.

Response
Correct the remote file name, read the file in from tape or restore the send file.
Repeat the command.

FTR2156 OPENFT: Request (&00). Remote system: Home directory not found (&01)

FTR2157 OPENFT: Request (&00). Remote system: Renaming not possible (&01)

FTR2158 OPENFT: Request (&00). Remote system: Not enough space for '(&01).

Meaning
The command was not executed (any further) because the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up. The receive file is no longer created/extended after the problem has occurred.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– increase the receive file's primary/secondary allocation.
If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2159 OPENFT: Request (&00). Remote system: File owner unknown.

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined on the relevant system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2160 OPENFT: Request (&00). Remote system: Invalid file password.

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Meaning
Correct the password in the file description or the command. Repeat the command.

FTR2161 OPENFT: Request (&00). Remote system: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2162 OPENFT: Request (&00). Remote system: File/directory '(&01)' is read only.

Meaning
The file or directory is write-protected.

Response
Correct the remote file name or remove the write protection of the remote file.
Repeat the command.

FTR2163 OPENFT: Request (&00). Remote system: File structure not supported(&01).

Meaning
The request cannot be carried out because the file structure is not supported. For example, an attempt was made to get a PLAM library or ISAM file from the BS2000 system.

Response
Transfer the file transparently.

FTR2164 OPENFT: Request (&00). Remote system: Syntax error in resulting file name(&01).

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected.
Possible reasons:
– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Meaning
Repeat the command using the correct syntax.

FTR2165 OPENFT: Request (&00). Remote system: Transparent file transfer not supported.

Meaning
The request could not be carried out because the partner system does not support the transfer of files in a transparent format.

FTR2166 OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2167 OPENFT: Request (&00). Remote system: Access to '(&01)' denied (&02).

Meaning
The command was not executed because the remote file only permits certain access modes.

Response
Correct the transfer direction, file name or file protection attributes on the remote system. Repeat the command.

FTR2168 OPENFT: Request (&00). Remote system: Follow-up processing exceeds length limit.

Meaning
The maximum length of follow-up processing was exceeded; see the command syntax description.

Response
Shorten the follow-up processing, or use procedures. Repeat the command.

FTR2169 OPENFT: Request (&00). Remote system: Transfer admission invalid.

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands are incorrect or the request was rejected by FTAC because of insufficient authorization.

Response
Define the requisite TRANSFER-ADMISSION or correct it or check the authorization entered in FTAC. Repeat the command if necessary.

FTR2170 OPENFT: Request (&00). Remote system: Function not supported (&01).

FTR2171 OPENFT: Request (&00). Remote system: Processing admission invalid.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands are incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary..

FTR2172 OPENFT: Request (&00). Remote system: Request queue full.

Meaning
The command was not executed because the maximum number of permissible file transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR2173 OPENFT: Request (&00). Remote system: User data encryption is mandatory

Meaning
The remote system only accepts requests using data encryption.

Response
Repeat the request using data encryption.

FTR2195 OPENFT: Request (&00). Remote system: openFT is not longer authorized to execute requests for this user.

FTR2196 OPENFT: Request (&00) has been canceled in the remote system.

Meaning
The request was deleted on the remote system before termination.

FTR2197 OPENFT: Request (&00). Remote system: File/directory '(&01)' could not be created(&02).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2198 OPENFT: Request (&00). Remote system: Higher-level directory no longer found

FTR2199 OPENFT: Request (&00). Remote system: I/O error for '(&01)' (&02).

Meaning
An error occurred at input/output. Possible cause:
– BS2000: DMS error, possibly the transfer ID.
– The send or receive files was deleted during transfer.
The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2200 OPENFT: Request (&00). Remote system: File now locked to prevent multiple access.

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously. An attempt is made, for example, to access a library opened in z/OS.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files not closed correctly. If a lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2201 OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer available(&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or because the file extends over more than one private disk or an attempt has been to transfer a file migrated by HSM.

Response
Inform the operator if necessary or carry out an HSM recall for the file.
Repeat the command.

FTR2202 OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer found.

Meaning
The command was not executed because the remote file is not or no longer in the catalog or on a volume of the corresponding system (e.g. after a restart).

Response
Restore the remote file. Repeat the command.

FTR2203 OPENFT: Request (&00). Remote system: Home directory no longer found (&01)

FTR2204 OPENFT: Request (&00). Remote system: File/directory '(&01)' gets no more space.

Meaning
The command was not executed (any further) because
– the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
– the send file contains too long a sequence of empty blocks, or
– the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can no longer be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.
If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2205 OPENFT: Request (&00). Remote system: File owner no longer known.

Meaning
The command was not executed because the owner of either the send file or the receive file is not defined on the relevant system, or because the file owner and the user requesting the creation of the receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2206 OPENFT: Request (&00). Remote system: Pre-/post-processing error (&01).

Meaning
The command executed in local pre-/postprocessing returned a result value other than OK.

Response
Correct the pre-/post-processing command and issue it again.

FTR2207 OPENFT: Request (&00). Remote system: Exit code (&01) during pre-/post-processing (&02).

Meaning
The command executed in local pre-/postprocessing returned the exit code (&01).

Response
Correct the pre-/post-processing command in accordance with the exit code and issue it again.

FTR2208 OPENFT: Request (&00). Remote system: File password no longer valid.

Meaning
The command was not executed because the password for the send file or receive file is missing or incorrect.

Response
Correct the password in the file description or the command. Repeat the command.

FTR2209 OPENFT: Request (&00). Remote system: File/directory '(&01)' is now read only.

FTR2210 OPENFT: Request (&00). Remote system: File structure error (&01).

Meaning
The command was not executed due to a file structure error.
File structure errors include:
– The attributes of the send file are incomplete.
– The data of the send file is incompatible with its structure attributes.
– The records of the send file are too long.
– If WRITE-MODE=EXTEND-FILE or the -e parameter are specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
– BS2000: The send or receive file is a member of an old LMS library (not PLAM).
– BS2000: The send file has an odd block factor (e.g. BLKSIZE=(STD,1)), and the receive file is stored on an NK4 pubset.

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE is specified, restore the receive file. Repeat the command.

FTR2211 OPENFT: Request (&00). Remote system: NDMS error (&01).

Response
Repeat the request a little later.

FTR2212 OPENFT: Request (&00). Recovery failed (&01).

Meaning
The restart could not be carried out. It may not have been possible to complete restart-capable pre-/post-processing before termination of the server process (waiting time: max. minutes).

Response
Repeat the command.

FTR2213 OPENFT: Request (&00). Remote system: Resource bottleneck.

Meaning
The order was rejected because the partner system currently does not have the resources available to accept requests. It is possible that the maximum number of concurrent connections in the partner system does not permit any additional connection.

Response
Repeat the request a little later. Where necessary, ask the administrator of the partner system to increase the maximum number of concurrent connections on their system.

FTR2214 OPENFT: Request (&00). Remote system: Access to '(&01)' is no longer permissible(&02).

Meaning
The command was not executed because
– the send file or receive file only permits certain access modes (e.g. read only) or a directory was specified as the source or destination of a file transfer.
– or because no valid password for an FTAC profile has been stored in the local system for executing the ftexec command from a remote system.

Response
Correct the transfer direction, write mode, file name or file protection attributes or specify a valid password for the FTAC profile. Repeat the command.

FTR2216 OPENFT: Request (&00). Remote system: File structure not supported (&01).

Meaning
The request cannot be carried out because the file structure is not supported. An attempt was made, for example, to get a PLAM library or ISAM file from BS2000.

Response
Transfer the file transparently.

FTR2217 OPENFT: Request (&00). Remote system: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2218 OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2225 OPENFT: Information output canceled.

Meaning
A show command was interrupted, for example.

Response
Repeat the command.

8.2.2 FTC messages

FTC0001 FTAC VERSION (&OO) ACTIVE

Meaning
FTAC initialization is concluded.

FTC0003 (&OO) LOGGING RECORDS DELETED

Meaning
The specified number of records have been deleted from the logging file.

FTC0050 CMD ACCEPTED. WARNING: LOWER ADM-LEVEL REMAINS IN EFFECT

Meaning
The set security level exceeds the administrator's limit value and will remain without effect until the administrator's limit value is increased.

Response
Request a higher maximum security level from the FTAC administrator.

FTC0051 CMD ACCEPTED. WARNING: TRANSFER-ADMISSION EXISTS AS USER ID

Meaning
A user ID with the same name already exists in the system.

Response
The message is simply intended to indicate a possible confusion.

FTC0052 CMD TERMINATED. INFORMATION INCOMPLETE

Meaning
Information output has been interrupted.

Response
Repeat the command if necessary.

FTC0053 CMD TERMINATED. NO FT PROFILE FOUND

Meaning
There is no FT profile for the specified criteria.

FTC0054 CMD ACCEPTED. NO INFORMATION AVAILABLE

Meaning
There is no information on the specified criteria.

FTC0055 WARNING: PARTNER RESTRICTION DOES NOT LONGER EXIST

FTC0056 WARNING: TRANSFER ADMISSION LOCKED

FTC0057 WARNING: ATTRIBUTES OF TRANSFER ADMISSION ARE IGNORED

Meaning
In the case of a profile with transfer admission *NOT-SPECIFIED, VALID, USAGE and EXPIRATION-DATE are ignored.

FTC0070 CMD TERMINATED. SHORTAGE OF RESOURCES

Meaning
The command cannot be executed due to a lack of resources.

Response
Repeat the command.

FTC0071 CMD REJECTED. OPENFT NOT ACTIVE

Meaning
openFT has not been activated, FTAC is therefore inactive.

Response
Ask the system administrator to activate openFT. FTAC will be activated by openFT.

FTC0100 CMD REJECTED. FT PROFILE ALREADY EXISTS

Meaning
There is already an FT profile with the specified name.

Response
Select another name.

FTC0101 CMD REJECTED. TRANSFER ADMISSION ALREADY EXISTS

Meaning
There is already an FT profile with the specified transfer admission.

Response
You should choose the TRANSFER-ADMISSION more carefully to ensure greater security.

FTC0102 FILE ALREADY EXISTS

FTC0103 INVALID FILE CONTENT OR ACCESS TO FILE DENIED

Meaning
The file is not an FTAC export file or access is prohibited.

FTC0104 ACCESS TO USER ID DENIED OR USER ID DOES NOT EXIST.

Meaning
Access to user ID rejected.
The user ID does not exist.

FTC0105 ACCESS TO FILE DENIED

FTC0106 ACCESS TO TEMPORARY FILE DENIED

FTC0107 NO SPACE AVAILABLE

FTC0108 THE VERSION OF EXPORT FILE IS NOT COMPATIBLE WITH CURRENT VERSION

FTC0109 FILE IS NO FTAC EXPORT FILE

FTC0110 FILE NAME TOO LONG

FTC0111 SYNTAX ERROR IN FILE NAME

FTC0112 CMD REJECTED. EXPIRATION DATE NOT VALID

Meaning
The value of the parameter EXPIRATION-DATE must be between 1970-01-02 and 2038-01-19.

FTC0150 CMD REJECTED. USER NOT AUTHORIZED FOR FTAC COMMANDS

Meaning
There is no password for the admission.

Response
Specify the FTAC password.

FTC0151 CMD REJECTED. USER NOT AUTHORIZED FOR THIS MODIFICATION

Meaning
Only the administrator or owner can perform the modification.

FTC0152 CMD REJECTED. USER NOT AUTHORIZED FOR OTHER USER IDS

Meaning
The specified user ID is not your own user ID.

FTC0153 CMD REJECTED. USER NOT AUTHORIZED FOR OTHER OWNER IDS

Meaning
The specified owner identification is not your own user ID.

FTC0154 CMD REJECTED. NO AUTHORIZATION FOR DELETION OF LOG RECORDS

FTC0155 CMD REJECTED. USER NOT AUTHORIZED FOR DIAGNOSE

Meaning
Only the FT administrator and FTAC administrator may call the diagnostic function.

FTC0156 COMMAND ALLOWED FOR FTAC ADMINISTRATOR ONLY

FTC0157 CMD REJECTED. NO AUTHORIZATION FOR THIS SET OF PARAMETERS

Meaning
The FTAC administrator can only create profiles with a transfer admission specification if he or she knows the complete user ID or possesses the SU privilege.

Response
Specify the full user ID in the form user-adm=(uid,acc,pw).

FTC0170 CMD REJECTED. GIVEN PARTNER UNKNOWN

Meaning
The specified partner is unknown within the group of partner systems permitted for this user.

FTC0171 CMD REJECTED. GIVEN FT PROFILE NAME UNKNOWN

Meaning
The specified profile does not exist.

FTC0172 CMD REJECTED. INVALID USER ADMISSION

Meaning
The specified user admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD is incorrect.

FTC0173 CMD REJECTED. INVALID PROCESSING ADMISSION

Meaning
The specified processing admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD specification is incorrect.

FTC0174 CMD REJECTED. MODIFICATION INVALID FOR NOT UNIQUE SELECTION CRITERIA

Meaning
The parameters "NEW-NAME" and "TRANSFER-ADMISSION" may only be used in combination with unique selection criteria ("NAME" or "TRANSFER-ADMISSION").

Response
Choose a unique selection criterion.

FTC0175 CMD REJECTED. MODIFICATION INVALID FOR STANDARD AUTHORIZATION RECORD

Meaning
The parameter "NEW-PASSWORD" may not be specified for *STD.

FTC0176 CMD REJECTED. GIVEN USER ID UNKNOWN

Meaning
The specified user ID does not exist in the system.

FTC0177 FILE UNKNOWN

FTC0178 MULTIPLE PARTNER NAME SPECIFIED

FTC0179 VIOLATION OF MAXIMAL NUMBER OF PARTNER RESTRICTIONS

FTC0180 MULTIPLE USERID SPECIFIED

FTC0181 MULTIPLE FT PROFILE NAME SPECIFIED

FTC0182 TOTAL MAXIMUM PARTNER NAME LENGTH EXCEEDED

Meaning
The total length of the partner names may not exceed 1000 characters.

FTC0183 CMD REJECTED. PARTNER NOT SUPPORTED

FTC0184 Invalid parameter transfer admission for profile *STD

Meaning
The transfer admission of the default profile must be *NOT-SPECIFIED.

FTC0185 COMBINATION OF THESE TRANSFER FUNCTIONS NOT ALLOWED

FTC0200 CMD REJECTED. FOLLOW-UP PROCESSING TOO LONG

Meaning
The total length of the two follow-up processing commands is too great.

Response
Use shorter commands (e.g. by using procedures).

FTC0201 USER ID TOO LONG

FTC0202 PROFILE NAME TOO LONG

FTC0203 TRANSFER ADMISSION TOO LONG

FTC0204 PARTNER TOO LONG

FTC0205 FULLY QUALIFIED FILE NAME TOO LONG

FTC0206 PARTIALLY QUALIFIED FILE NAME TOO LONG

FTC0207 PROCESSING COMMAND TOO LONG

FTC0208 INVALID DATE SPECIFIED

FTC0209 INVALID TIME SPECIFIED

FTC0210 TRANSFER ADMISSION TOO SHORT

FTC0211 PARAMETERS (&00) AND (&01) MAY NOT BE SPECIFIED TOGETHER

FTC0212 LICENSE CHECK ERROR (&00) FOR FTAC

FTC0213 MANDATORY PARAMETER PROFILE NAME IS MISSING

FTC0214 MANDATORY PARAMETER FILE NAME IS MISSING

FTC0215 SYNTAX ERROR IN PARAMETER (&OO)

FTC0216 PASSWORD TOO LONG

FTC0217 TEXT TOO LONG

FTC0218 TOO MANY PARTNERS

FTC0219 TOO MANY USERS

FTC0220 TOO MANY PROFILES

FTC0250 LOAD ERROR. ERROR-CODE (&OO)

FTC0251 CMD REJECTED. FTAC NOT AVAILABLE

Meaning
openFT-AC has not been installed completely.

Response
The system administrator must check the openFT-AC installation.

FTC0253 FTAC COMMAND NOT FOUND IN SYNTAXFILE

Meaning
The openFT-AC syntax file has been merged incorrectly or incompletely into the system syntax file.

Response
The system administrator must check the system syntax file.

FTC0254 SYSTEM ERROR. ERRORCODE (&OO)

Meaning
A system error has occurred.

Response
Generate diagnostic material and inform the staff responsible for system diagnostics.

FTC0255 CMD TERMINATED. SYSTEM ERROR

Meaning
A system error has occurred.

Response
Inform the system administrator. At the same time a message is issued to the operator terminal providing exact troubleshooting information.

FTC1001 SUBMISSION REJECTED. INVALID TRANSFER-ADMISSION

Meaning
The specified TRANSFER-ADMISSION is not defined in any FT profile.

FTC1002 SUBMISSION REJECTED. INVALID INITIATOR

Meaning
The FT profile restricts initiatives to LOCAL or REMOTE.

FTC1003 SUBMISSION REJECTED. INVALID TRANSFER-DIRECTION

Meaning
The FT profile restricts the TRANSFER-DIRECTION to TO or FROM.

FTC1004 SUBMISSION REJECTED. INVALID PARTNER NAME

Meaning
The FT profile does not permit any requests involving the specified partner system.

FTC1005 SUBMISSION REJECTED. VIOLATION OF MAX-PARTNER-LEVEL

Meaning
The partner system's security level exceeds the value specified for MAX-PARTNER-LEVEL in the FT profile.

FTC1006 SUBMISSION REJECTED. SYNTAX ERROR OF FILE NAME EXPANSION

Meaning
The FT profile does not permit the specification of a file name or file name expansion in the request.

FTC1007 SUBMISSION REJECTED. VIOLATION OF LIBRARY RESTRICTION

Meaning
The file or library name specified in the command infringes the LIBRARY restriction in the profile.

FTC1008 SUBMISSION REJECTED. VIOLATION OF ELEMENT RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT in the request.

FTC1009 SUBMISSION REJECTED. VIOLATION OF ELEMENT-VERSION RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT-VERSION in the request.

FTC100A SUBMISSION REJECTED. VIOLATION OF ELEMENT-TYPE RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT-TYPE in the request.

FTC100B SUBMISSION REJECTED. VIOLATION OF FILE-PASSWORD RESTRICTION

Meaning
The FT profile does not permit the specification FILE-PASSWORD in the request.

FTC100C SUBMISSION REJECTED. VIOLATION OF USER-IDENTIFICATION(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification USER-IDENTIFICATION in the request's PROCESSING-ADMISSION.

FTC100D SUBMISSION REJECTED. VIOLATION OF ACCOUNT(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification ACCOUNT in the request's PROCESSING-ADMISSION.

FTC100E SUBMISSION REJECTED. VIOLATION OF PASSWORD(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification PASSWORD in the request's PROCESSING-ADMISSION.

FTC100F SUBMISSION REJECTED. VIOLATION OF SUCCESS-PROCESSING RESTRICTION

Meaning
The FT profile does not permit the specification SUCCESS-PROCESSING.

FTC1010 SUBMISSION REJECTED. VIOLATION OF FAILURE-PROCESSING RESTRICTION

Meaning
The FT profile does not permit the specification FAILURE-PROCESSING.

FTC1011 SUBMISSION REJECTED. VIOLATION OF WRITE-MODE RESTRICTION

Meaning
The FT profile does not permit the specified WRITE-MODE.

FTC1012 SUBMISSION REJECTED. INVALID FT-FUNCTION

Meaning
The FT profile does not permit the desired FT function.

FTC1013 SUBMISSION REJECTED. VIOLATION OF PROFILE WITH CHIPCARD-ID

Meaning
The profile may only be used with a chipcard.

FTC1014 SUBMISSION REJECTED. VIOLATION OF DATA ENCRYPTION RESTRICTION

Meaning
The profile does not permit the value DATA-ENCRYPTION in the request.

FTC2001 SUBMISSION REJECTED. SYNTAX ERROR ON FILE NAME EXPANSION

Meaning
The combination of the FT profile's FILE-NAME and FILE-NAME expansion resulted in a syntax error.

FTC2002 SUBMISSION REJECTED. SYNTAX ERROR ON LIBRARY NAME EXPANSION

Meaning
The combination of the FT profile's LIBRARY name and LIBRARY expansion resulted in a syntax error.

FTC2003 SUBMISSION REJECTED. SYNTAX ERROR ON ELEMENT NAME EXPANSION

Meaning
The combination of the FT profile's ELEMENT name and ELEMENT expansion resulted in a syntax error.

FTC2004 SUBMISSION REJECTED. TOTAL LENGTH OF RESULT PROCESSING EXCEEDS 500 CHARACTERS

Meaning
SUCCESS and FAILURE processing including the expansions defined in the FT profile exceeds 1000 characters.

FTC3001 SUBMISSION REJECTED. INVALID USER-IDENTIFICATION

Meaning
The TRANSFER-ADMISSION's USER-IDENTIFICATION or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3002 SUBMISSION REJECTED. INVALID ACCOUNT

Meaning
The TRANSFER-ADMISSION's ACCOUNT specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3003 SUBMISSION REJECTED. INVALID PASSWORD

Meaning
The TRANSFER-ADMISSION's PASSWORD specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3004 SUBMISSION REJECTED. TRANSFER ADMISSION LOCKED

Meaning
The transfer admission is locked. The reasons may be ascertained from the output from the FTSHWPRF command.

FTC3011 SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the user for the OUTBOUND SEND function class.

FTC3012 SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the user for the OUTBOUND RECEIVE function class.

FTC3013 SUBMISSION REJECTED. VIOLATION OF USER INBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND SEND function class.

FTC3014 SUBMISSION REJECTED. VIOLATION OF USER INBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND RECEIVE function class.

FTC3015 SUBMISSION REJECTED. VIOLATION OF USER INBOUND PROCESSING LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND PROCESSING function class.

FTC3016 SUBMISSION REJECTED. VIOLATION OF USER INBOUND FILE MANAGEMENT LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND FILE MANAGEMENT function class

FTC3021 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the OUTBOUND SEND function class.

FTC3022 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the OUTBOUND RECEIVE function class.

FTC3023 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND SEND function class.

FTC3024 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND RECEIVE function class.

FTC3025 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND PROCESSING LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND PROCESSING function class.

FTC3026 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND FILE MANAGEMENT LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND FILE MANAGEMENT function class

8.3 Using openFT in z/OS systems without the TSO interactive system

openFT is intended for use under the z/OS operating system. The commands are passed to the TSO command processor. Nevertheless, openFT can also be used without the TSO interactive system. In this case, the IBM utility IKJEFT01 must be used to call the TSO command processor in batch mode.

In order to be able to work with openFT without the TSO interactive system, all commands must be included in batch jobs. These jobs are initiated via the IBM utility IEBGENER. IEBGENER reads the job information from a file and passes it on to the Job Entry Subsystem (JES2/3).

Issuing TSO commands

These commands are processed by the TSO command processor. In an exclusive z/OS batch environment, the IKJEFT01 utility provides the appropriate interface.

Example of a batch job including the NCOPY command:

```
//USERN      JOB    .....
//NCOPY      EXEC PGM=IKJEFT01
//SYSPRINT   DD   SYSOUT=*
//SYSTSPRT   DD   SYSOUT=*
//SYSTSIN    DD   *
NCOPY TRANS=TO,PARTNER=MVS2,+
LOC=(FILE=......
...
...
/*
//
```

Glossary

Italic type indicates a reference to other terms in this glossary.

ABEND
Abnormal termination of program.

access protection
Comprises all the methods used to protect a data processing system against unauthorized system access.

access right / access admission
Derived from the *transfer admission*. The access right defines the scope of access for the user who specifies the transfer admission.

ACF-2
Program product from Computer Associates for system and data access control.

ADM administrator
Administrator of the *remote administration server*. This is the only person permitted to modify the configuration data of the remote administration server.

ADM partner
Partner system of an openFT instance with which communication takes place over the *FTADM protocol* in order to perform *remote administration*.

ADM traps
Short messages sent to the *ADM trap server* if certain events occur during operation of openFT.

ADM trap server
Server that receives and permanently stores the *ADM traps*. It must be configured as a *remote administration server*.

administrated openFT instance
openFT instances that are able to be administered by *remote administrators* during live operation.

admission profile

Way of defining the *FTAC* protection functions. Admission profiles define a *transfer admission* that has to be specified in *FT requests* instead of the *LOGON* or *Login authorization*. The admission profile defines the *access rights* for a user ID by restricting the use of parameters in *FT requests*.

admission profile, privileged

see *privileged admission profile*

admission set

In *FTAC*, the admission set for a particular user ID defines which FT functions the user ID may use and for which *partner systems*.

admission set, privileged

see *privileged admission set*

AES (Advanced Encryption Standard)

The current symmetrical encryption standard, established by NIST (National Institute of Standards and Technology), based on the Rijndael algorithm, developed at the University of Leuven (B). The openFT product family uses the AES method to encrypt the request description data and possibly also the file contents.

alphanumeric

Alphanumeric characters comprise alphabetic and numeric characters, i.e. the letters A-Z and the digits 0-9 as well as the additional characters $, @, #.

AMODE

Specification for addressing a module (24-bit or 31-bit addresses).

ANSI code

Standardized 8-bit character code for message exchange. The acronym stands for "American National Standards Institute".

API (Application Programming Interface)

An interface that is freely available to application programmers. It provides a set of interface mechanisms designed to support specific functionalities.

asynchronous request

Once the *FT request* has been submitted, it is processed independently of the user. The user can continue working once the system has confirmed acceptance of the request. (see also *synchronous request*).

authentication

Process used by openFT to check the unique identity of the request partner.

basic functions

Most important file transfer functions. Several basic functions are defined in the *admission set* which can be used by a login name. The six basic functions are:

– inbound receive
– inbound send
– inbound follow-up processing
– inbound file management
– outbound receive
– outbound send

central administration

Central administration in openFT incorporates the *remote administration* and *ADM traps* functions and requires the use of a *remote administration server*.

Character Separated Values (CSV)

This is a quasi-tabular output format that is very widely used in the PC environment in which the individual fields are separated by a separator (often a semicolon ";"). It permits the further processing of the output from the most important openFT commands using separate tools.

client

– Term derived from client/server architectures: the partner that makes use of the services provided by a *server*.
– Logical instance which submits requests to a *server*.

cluster

A number of computers connected over a fast network and which in many cases can be seen as a single computer externally. The objective of clustering is generally to increase the computing capacity or availability in comparison with a single computer.

Comma Separated Values

see *Character Separated Values*.

communication computer

Computer for constructing a *data communication system*.

communication controller

see *preprocessor*

compression

This means that several identical successive characters can be reduced to one character and the number of characters is added to this. This reduces transfer times.

computer network, open

see *open computer network*

connectivity

In general, the ability of systems and partners to communicate with one another. Sometimes refers simply to the communication possibilities between transport systems.

cross domain connection

A connection between computers that are located in different SNA domains. A cross domain connection from a TRANSDATA network to an SNA network requires the software product TRANSIT-CD to be used as a gateway.

cross network connection

A connection between computer that are located in different SNA networks. A cross network connection from a TRANSDATA network to one or more SNA networks requires the software product TRANSIT-CD and, depending on the configuration, may also require TRANSIT-SNI to be used as a *gateway*.

data communication system

Sum of the hardware and software mechanisms which allow two or more communication partners to exchange data while adhering to specific rules.

data compression

Reducing the amount of data by means of compressed representation.

data encoding

Way in which an *FT system* represents characters internally.

Data Encryption Standard (DES)

International data encryption standard for improved security. The DES procedure is used in the FT products to encrypt the request description data and possibly the request data if connections are established to older versions of openFT that do not support *AES*.

data protection

- In the narrow sense as laid down by law, the task of protecting personal data against misuse during processing in order to prevent the disclosure or misappropriation of personal information.
- In the wider sense, the task of protecting data throughout the various stages of processing in order to prevent the disclosure or misappropriation of information relating to oneself or third parties.

data security

Technical and organizational task responsible for guaranteeing the security of data stores and data processing sequences, intended in particular to ensure that

- only authorized personnel can access the data,
- no undesired or unauthorized processing of the data is performed,
- the data is not tampered with during processing,
- the data is reproducible.

data set

File.

DHCP

Service in TCP/IP networks that automatically assigns IP addresses and TCP/IP parameters to clients on request.

Direct Access Storage Device (DASD)

Disk storage device.

directory

Directories are folders in the hierarchical file system of a Unix system (including POSIX) or a Windows system that can contain files and/or further directories. openFT for z/OS interprets, on the one hand, the contents of a PO or PDSE data set (and the members included in it) as a directory, and on the other hand also all files with a common name up to a qualifying delimiter (dot).

dynamic partner

partner system that is either not entered in the *partner list* (*free dynamic partner*) or that is entered in the partner list with only address but without a name (*registered dynamic partner*).

emulation

Components that mimic the properties of another device.

Explorer

A program from Microsoft that is supplied with Windows operating systems to facilitate navigation within the file system.

file attributes

A file's properties, for example the size of the file, access rights to the file or the file's record structure.

file management

Possibility of managing files in the remote system. The following actions are possible:

- Create directories
- Display and modify directories
- Delete directories
- Display and modify file attributes
- Rename files
- Delete files.

file processing

The openFT "file processing" function makes it possible to send a receive request in which the output of a remote command or program is transferred instead of a remote file.

file transfer request

see *FT- request*

firewall processor

Processor which connects two networks. The possible access can be controlled precisely and also logged.

fixed-length record

A record in a file all of whose records possess the same, agreed length. It is not necessary to indicate this length within the file.

follow-up processing

FT function that initiates execution of user-specified commands or statements in the *local* and/or the *remote system* after an *FT request* has been completed. The user may define different follow-up processing, depending on the success or failure of FT request processing. See also *preprocessing* and *postprocessing*.

follow-up processing request

Statements contained within an *FT request* which perform *follow-up processing* after file transfer.

free dynamic partner
Partner system that is not entered in the partner list.

FT administrator
Person who administers the openFT product installed on a computer, i.e. who is responsible, among other things, for the entries in the *network description file* or the *partner list* as well as for controlling resources.

FT request
Request to an *FT system* to transfer a file from a *sending system* to a *receive system* and (optionally) start *follow-up processing requests*.

FT system
System for transferring files that consists of a computer and the software required for file transfer.

FT trace
Diagnostic function that logs FT operation.

FTAC (File Transfer Access Control)
Extended access control for file transfer and file management. In the case of BS2000 and z/OS, this is implemented by means of the product openFT-AC, for other operating systems it is a component of the openFT product, e.g. in openFT for Unix systems or openFT for Windows systems.

FTAC administrator
Person who manages openFT-AC on a computer.
The FTAC administrator specifies for their system, among other things, the security-technical framework in the form of a standard admission set that is valid for all users.
In z/OS the FTAC administrator is also responsible for managing admission sets and authorization profiles.

FTAC logging function
Function which FTAC uses to log each access to the protected system via file transfer.

FTADM protocol
Protocol used for communication between two openFT instances in order to perform *remote administration* or transfer *ADM traps*.

FTAM protocol (File Transfer, Access and Management)
Protocol for file transfer standardized by the “International Organization for Standardization” (ISO) (ISO 8571, FTAM).

FTP partner

Partner system that uses *FTAM protocols* for communication.

FTP protocol

Manufacturer-independent protocol for file transfer in TCP/IP networks.

gateway

Generally understood to mean a computer that connects two or more networks and which does not function as a bridge. Variants: gateway at network level (= router or OSI relay), transport and application gateway.

gateway processor

Communication computer that links a computer network to another computer network. The mapping of the different protocols of the various computer networks takes place in gateway processors.

Generalized Trace Facility (GTF)

IBM tool for generating traces (in particular for monitoring the data traffic between an application program and the relevant VTAM applications and between VTAM applications and the data communication line).
global request identification / global request ID Request number that the *initiator* of an openFT or FTAM request transfers to the *responder*. This means that the global request ID in the responder is identical to the *request ID* in the initiator. The responder generates its own (local) request ID for the request. This means that information stored in both the initiator and the responder can be unambiguously assigned to a request. This is particularly important if the request has to be restarted.

heterogeneous network

A network consisting of multiple subnetworks functioning on the basis of different technical principles.

homogeneous network

A network constructed on the basis of a single technical principle.

identification

Procedure making it possible to identify a person or object.

IEBCOPY

IBM tool for copying libraries (PO or PDSE data sets).

IEBGENER

IBM tool for copying sequential files (PS data sets).

IEBPTPCH
IBM tool for printing files.

inbound file management
Request issued in a *remote system* for which directories or file attributes of the *local system* can be displayed, file attribute modified or local file deleted.

inbound follow-up processing
Request issued in a *remote system* with *follow-up processing* in the *local system*.

inbound receive
Request issued in the *remote system*, for which a file is received in the *local system*.

inbound request / inbound submission
Request issued in another system, i.e. for this request.

inbound send
Request issued in a *remote system* for which a file is sent from the *local system* to the remote system.

initiator
Here: *FT system* that submits an *FT request*.

instance / entity
A concept of OSI architecture: active element in a layer. Also see *openFT instance*.

instance ID
A network-wide, unique address of an openFT instance.

integrity
Unfalsified, correct data following the processing, transfer and storage phases.

Interactive Problem Control System (IPCS)
IBM tool for formatting a machine-readable (unformatted) dump.

interoperability
Capability of two *FT systems* to work together.

ISO/OSI reference model
The ISO/OSI Reference Model is a framework for the standardization of communications between open systems. (ISO=International Standards Organization).

ISPF, ISPF/PDF
Menu-driven utilities for software development and for conducting a (TSO) dialog.

job
A sequence of JCL statements (batch).

job transfer
Transfer of a file that constitutes a *job* in the *receive system* and is initiated as a job there.

library
File with internal structure (members)

library member
Part of a library. A library member may in turn be subdivided into a number of records.

Local Area Network (LAN)
Originally a high-speed network with limited physical extension. Nowadays, any network, that uses CSMA/CD, Token Ring or FDDI irrespective of the range (see also *WAN Wide Area Network*).

local system
The *FT system* at which the user is working.

logging function
Function used by openFT to log all file transfer accesses to the protected system.

log record
Contains information about access checks performed by openFT (FTAC log record) or about a file transfer or remote administration request which is started when the access check was successful (FT log record or ADM log record).

Logical Unit (LU)
Interface between an application program and the SNA data communications network. The LU type describes the communications characteristics.

Login authorization
Transfer admission to a computer which (as a rule) consists of the login name and the password, and authorizes dialog operation, see also *LOGON authorization*.

LOGON authorization

Transfer admission authorizing access to a computer. The LOGON authorization (normally) consists of user ID, account number and password and authorizes the user to make use of interactive operation.

mainframe

Computer (consisting of one or more processors) which runs under the control of a universal operating system (e.g. BS2000 or z/OS).
Synonyms: BS2000 computer, host computer.

named partner

partner system entered by its name in the *partner list*.

Network Control Program (NCP)

Operating system of the front-end-processor for SNA hosts.

NetMaster

Tool for controlling a data communication system.

NetView

IBM tool for controlling a data communication system.

network description file

File used up to openFT V9 that contains specifications concerning *remote systems* (*FT systems*).

object

Passive element in a DP system that contains or receives data and which can be the object of an operation such as read, write or execute etc.
Examples: files, user IDs

offline logging

The log file can be changed during operation. Following this changeover, the previous log file is retained as an offline log file; new log records are written to a new log file. It is still possible to view the log records in an offline log file using the tools provided by openFT.

open computer network

Computer network in which communication is governed by the rules of ISO/OSI. Interoperation of different computers from various vendors is made possible by defined *protocols*.

openFT instance

Several openFT systems, so-called openFT instances, can be running simultaneously on an individual computer or on a Sysplex cluster. Each instance has its own address (instance ID,host) and is comprised of the loaded code of the openFT products (including add-on products if they are available) and of the variable files such as the network description file or partner list, logging files, key library, request queue, etc.

openFT partner

Partner system which is communicated with using *openFT protocols*.

openFT protocols

Standardized *protocols* for file transfer (SN77309, SN77312).

openFT-FTAM

Add-on product for openFT (for BS2000, Unix systems and Windows systems) that supports file transfer using FTAM protocols. FTAM stands for File Transfer, Access and Management (ISO 8571).

operating parameters

Parameters that control the *resources* (e.g. the permissible number of connections).

outbound request / outbound submission

Request issued in your own processor.

outbound receive

Request issued locally for which a file is received in the *local system*.

outbound send

Request issued locally for which a file is sent from the *local system*.

owner of an FT request

User ID in the *local system* or *remote system* under which the *FT request* is started (or submitted):

– The owner of an FT request submitted on the local system is the user ID under which the request was issued.
– The owner of an FT request submitted on a remote system is the user ID accessed for the request on the local system (TRANSFER-ADMISSION).

partitioned data set extended (PDSE data set)
Library in the IBM z/OS Data Management System. Contains individual members and can be used instead of a partitioned organized data set. The IBM software product "Data Facility Storage Management Subsystem" (DFSMS) is required to use PDSE.

partitioned organized data set (PO data set)
Library of the IBM z/OS Data Management System. Contains individual members.

partner
see *partner system*

partner list
File containing specifications concerning *remote systems* (*FT systems*).

partner system
Here: *FT system* that carries out *FT requests* in cooperation with the *local system*.

password
Sequence of characters that a user must enter in order to access a user ID, file, job variable, network node or application. The user ID password serves for user *authentication*. It is used for access control. The file password is used to check access rights when users access a file (or job variable). It is used for file protection purposes.

physical sequential data set / PS data set
Sequential file in the IBM z/OS Data Management System; similar to a BS2000 SAM file.

Physical Unit (PU)
Each node of an SNA network contains a Physical Unit (PU) as an addressable instance. This is responsible for monitoring the connection to the host and for monitoring the *Logical Units* (LUs).

port number
Number that uniquely identifies a TCP/IP application or the end point of a TCP/IP connection within a processor.

POSIX (Portable Open System Interface)
Board and standards laid down by it for interfaces that can be ported to different system platforms.

postprocessing

openFT makes it possible to process the received data in the receiving system through a series of operating system commands. Postprocessing runs under the process control of openFT (in contrast to *follow-up processing*).

preprocessing

The preprocessing facility in openFT can be used to send a receive request in which the outputs of a remote command or program are transferred instead of a file. This makes it possible to query a database on a remote system, for example. Preprocessing also may be issued locally.

preprocessor / communication controller

A processor system connected upstream of the mainframe which performs special communication tasks in the network. Synonym: communication processor.

private key

Secret decryption key used by the recipient to decrypt a message that was encrypted using a *public key*. Used by a variety of encryption procedures including the *RSA procedure*.

privileged admission profile

Admission profile that allows the user to exceed the *FTAC administrator's* presettings in the *admission set*. This must be approved by the *FTAC administrator* who is the only person able to privilege admission profiles.

privileged admission set

Admission set belonging to the *FTAC administrator*.

procedure

Here: command procedure, corresponds in principle to an IBM CLIST or REXX procedure.

profile

In OSI, a profile is a standard which defines which protocols may be used for any given purpose and specifies the required values of parameters and options. Here: a set of commands assigned to a user ID. The permissibility of these commands is ensured by means of syntax files. See also *admission profile*, *privileged admission profile*.

protocol

Set of rules governing information exchange between peer partners in order to achieve a defined objective. This usually consists of a definition of the messages that are to be exchanged and the correct sequencing of messages including the handling of errors and other exceptions.

public key

Public encryption key defined by the receiver of a message, and made public or made known to the sender of the message. This allows the sender to encrypt messages to be sent to the receiver. Public keys are used by various encryption methods, including the *Rivest Shamir Adleman (RSA) procedure*. The public key must match the *private key* known only to the receiver.

RACF

IBM product for system and data access control.

receive file

File in the *receive system* in which the data from the *send file* is stored.

receive system

System to which a file is sent. This may be the *local system* or the *remote system*.

record

Set of data that is treated as a single logical unit.

registered dynamic partner

Partner system that is entered in the partner list with only an address but no name.

relay program

Program in a *gateway processor* that maps the different protocols onto one another.

remote administration

Administration of openFT instances from remote computers.

remote administration server

Central component required for *remote administration* and for *ADM traps*. A remote administration server runs on a Unix or Windows system running openFT as of V11.0. If it is used for *remote administration*, it contains all the configuration data required for this purpose.

remote administrator
Role configured on the *remote administration server* and which grants permission to execute certain administration functions on certain openFT instances.

remote system
see *partner system*

request
see *FT request*

request queue
File containing *asynchronous requests* and their processing statuses.

request identification / request ID
The (serial) number assigned to the request by the local system. In some commands, users are able to identify the request on the basis of this number. Here: Number assigned by the local system that identifies an *FT request.*

request management
FT function responsible for managing *FT requests*; it ensures request processing from the submission of a request until its complete processing or termination.

request number
see *request identification*

request storage
FT function responsible for storing *FT requests* until they have been fully processed or terminated.

resources
Hardware and software components needed by the *FT system* to execute an *FT request* (, connections, lines). These resources are controlled by the *operating parameters*.

responder
Here: *FT system* addressed by the *initiator*.

restart
Automatic continuation of an *FT request* following an interruption.

restart point
Point up to which the data of the *send file* has been stored in the *receive file* when a file transfer is interrupted and at which the transfer of data is resumed following a *restart*.

result list[ing]

List with information on a completed file transfer. This is supplied to the user in the *local system* and contains information on his or her *FT requests*.

REXX

IBM procedure language.

RFC (Request for Comments)

Procedure used on the Internet for commenting on proposed standards, definitions or reports. Also used to designate a document approved in this way.

RFC1006

Supplementary protocol for the implementation of ISO transport services (transport class 0) using TCP/IP.

Rivest-Shamir-Adleman-procedure (RSA procedure)

Encryption procedure named after its inventors that operates with a key pair consisting of a *public key* and a *private key*. Used by the openFT product family in order to reliably check the identity of the partner system and to transmit the AES key to the partner system for encrypting the file contents.

Secure FTP

Method by which a connection is tunneled using the *FTP protocol*, thus allowing secure connections with encryption and *authentication*.

security level

When FTAC is used, the security level indicates the required level of protection against a *partner system*.

send file

File in the *sending system* from which data is transferred to the *receive file*.

sending system

Here: *FT system* that sends a file. This may be the *local system* or the *remote system*.

server

Logical entity or application component which executes a client's requests and assures the (coordinated) usage of all the generally available services (File, Print, data base, Communication, etc.). May itself be the client of another server.

service

- As used in the OSI architecture: a service is the set of functions that a service provider makes available at a service access point.
- As used in the client/server architecture: a set of functions that a server makes available to its clients.
- Term used in Unix and Windows systems: A program, routine or process used to perform a particular system function to support other programs, in particular on a low level (hardware-related).

session

- In OSI, the term used for a layer 5 connection.
- In SNA, a general term for a connection between communication partners (applications, devices or users).

session selector

Subaddress used to address a *session* application.

SMF (System Management Facility)

IBM tool for collecting accounting data and statistics.

SMP/E (System Modification Program/Extended)

IBM product used to install and manage the software products, their versions and corrections.

SNA network

Data communication system that implements the Systems Network Architecture (SNA) of IBM.

SNMP (Simple Network Management Protocol)

Protocol for TCP/IP networks defined by the Internet Engineering Task Force (IETF) for the transfer of management information.

standard admission set

This standard admission set applies by default to all users for whom there is no dedicated admission set. These default settings may be restricted further by the user for his or her own admission set.

string

Character string

SU Privilege

Privilege of an FTAC administrator in z/OS. This privilege allows the administrator to set up admission profiles for which TRANSFER-ADMISSIONS have been released on other user IDs without the need to know the current password. This privilege is defined in the FTACADM member of the parameter library.

synchronous request

The user task that submitted the *FT request* waits for transfer to terminate. The user cannot continue working (see also *asynchronous request*).

system

see *FT- system*

system, local

see *local system*

system, remote

see *remote system*

task

Entity responsible for executing one or more programs within a *job*.

TCP/IP (Transmission Control Protocol / Internet Protocol)

Widely used data transmission protocol (corresponds approximately to layers 3 and 4 of the *ISO/OSI reference model*, i.e. network and transport layers); originally developed for the ARPANET (computer network of the US Ministry of Defense) it has now become a de-facto standard.

Top Secret

Program authored by the company Computer Associates for data and system access control.

transfer admission

Authorization for file transfer and file management when using FTAC. The transfer admissions is then used in place of the *LOGON or LOGIN authorization*.

Transmission Control Protocol / Internet Protocol

see *TCP/IP*

transport connection

Logical connection between two users of the transport system (terminals or applications).

transport layer

Layer 4 of the *ISO/OSI reference model* on which the data transport protocols are handled.

transport protocol

Protocol used on the *transport layer*

transport selector (T-selector)

Subaddress used to address an ISO-8072 application in the *transport layer*.

transport system

- The part of a system or architecture that performs approximately the functions of the four lower OSI layers, i.e. the transport of messages between the two partners in a communication connection.
- Sum of the hardware and software mechanisms that allow data to be transported in computer networks.

Unicode

The universal character encoding, maintained by the Unicode Consortium. This encoding standard provides the basis for processing, storage and interchange of text data in any language in all modern software and information technology protocols. The Unicode Standard defines three Unicode encoding forms: UTF-8, UTF-16 and UTF-32.

UNIX®

Registered trademark of the Open Group for a widespread multiuser operating system. A system may only bear the name UNIX if it has been certified by the Open Group.

Unix system

Commonly used designation for an operating system that implements functions typical of UNIX® and provides corresponding interfaces. POSIX and Linux are also regarded as Unix systems.

user identification / user ID

A name with a maximum length of eight characters. The user ID identifies the user when accessing the system. All files are set up under a user ID. .

variable length record

A record in a file all of whose records may be of different lengths. The record length must either be specified in a record length field at the start of the record or must be implicitly distinguishable from the next record through the use of a separator (e.g. Carriage Return - Line Feed).

VSAM

IBM file access method for sequential, direct and indexed access.

VTAM

IBM telecommunication access method.

WAN (Wide Area Network)

A public or private network that can span large distances but which runs relatively slowly and with higher error rates when compared to a *LAN*. Nowadays, these definitions have only limited validity. Example: in ATM networks.

Abbreviations

ACF/NCP	Advanced Communications Function/Network Control Program
ACF/VTAM	Advanced Communications Function/Virtual Telecommunicatio Access Method
ACF-2	Access Control Facility 2
AMODE	addressing mode
APF	Authorized Program Facility
ASCII	American Standard Code for Information Interchange
CCS	Coded Character Set
CCSN	Coded Character Set Name
CPPL	Command Processor Parameter List
CSV	Comma Separated Value
DA	Direct Access (data set)
DAS	Data Access Service
DASD	Direct Access Storage Device
DCAM	Data Communication Access Method
DES	Data Encryption Standard
DFSMS	Data Facility Storage Management Subsystem
DMS	Data Management System
DNS	Domain Name System
DSCB	Data Set Control Block
EBCDIC	Extended Binary Coded Decimal Interchange Code
FJAM	File Job Access Method
FT	File Transfer
FTAC	File Transfer Access Control
FTAM	File Transfer, Access and Management
FTP	File Transfer Protocol
GTF	Generalized Trace Facility

HSM	Hierarchical Storage Manager
IPCS	Interactive Problem Control System
ISO	International Organization for Standardization
ISPF	Interactive System Productivity Facility
ISPF/PDF	ISPF/Program Development Facility
JCL	Job Control Language
JES	Job Entry Subsystem
Kb	Kilobyte
LAN	Local Area Network
LMS	Library Maintenance System
LU	Logical Unit
Mb	Megabyte
MVS	Multiple Virtual Storage
MVS/ESA	MVS/Enterprise System Architecture
MVS/SP	MVS/System Product
MVS/XA	MVS/Extended Architecture
NCP	Network Control Program
NPSI	NCP Packet Switching Interface
NDMS	Network Data Management System
OMVS	OpenEdition MVS
OSI	Open Systems Interconnection
PDF	Program Development Facility
PDN *	Program System for Teleprocessing and Network Control
PDS	Partitioned Data Set
PDSE	Partitioned Data Set Extended
PEM	Privacy Enhanced Mail
PKCS	Public Key Cryptography Standards
PO	Partitioned Organized (data set)
POSIX	Portable Open System Interface
PS	Physical Sequential (data set)
PSCB	Protected Step Control Block
PTF	Program Temporary Fix
PU	Physical Unit

PUT	Program Update Tape
RACF	Resource Access Control Facility
REXX	Restructured Extended Executor (language)
RFC	Request for Comments
RFC1006	Request for Comments 1006
RMODE	Residence Mode
RSA	Rivest, Shamir, Adleman
SAM	Sequential Access Method
SDSF	System Display and Search Facility
SMF	System Management Facility
SMP/E	System Modification Program/Extended
SNA	Systems Network Architecture
SSL	Secure Socket Layer
SVC	Supervisor Call
TCP/IP	Transmission Control Protocol/Internet Protocol
TSO	Time Sharing Option
TSO/E	TSO/Extension
TSS	Top-Secret
UPT	User Profile Table
VSAM	Virtual Storage Access Method
VSAM-ES	Virtual Storage Access Method - Entry Sequenced
VTAM	Virtual Telecommunication Access Method
WAN	Wide Area Network

* German abbreviation

Additional documentation

The available literature as well as current information about the openFT line of products can be found on the Internet under http://manuals.ts.fujitsu.com/
Here you will also find pdf copies of all manuals which you can download.

The appropriate documentation from IBM can be obtained on the Internet using your customer number in the usual manner.

Index

%ELEMNAME
variable 114
%ELEMTYP
variable 114
%ELEMVERS
variable 114
%FILENAME
variable 114, 117
%JOBCLASS
variable 114
%PARTNER
variable 114, 117
%PARTNERAT
variable 114
%RESULT
variable 114
%RESULT variable 117
%TEMPFILE 111, 329
*DELETE (follow-up processing) 115
*DIRECTORY
operand description (display log records) 265
*FILE-PROCESSING
operand description (modify profile) 238
*ftmonitor
file name prefix 179, 232
*LOCKED
request status 356
*MODIFY-FILE-ATTRIBUTES
operand description (modify profile) 238
*READ-DIRECTORY
operand description (modify profile) 238
*SUSPEND
request status 356
*TRANSFER-FILE
operand description (modify profile) 237
*WAIT
request status 356
<nummer 1..ffff>
operand description (display information on reason codes in the logging records) 202

A

abbreviate
commands 152
abbreviated forms 152
ABEND 465
abort file transfer
with time specification 341
access
to the remote system 166, 189, 199
access admission 465
access check
FTAC 41
access protection 38, 465
BS2000 106
Unix system 105
Windows 105
access right 465
access rights 106
ACCESS-MODE
operand description (modify remote file attributes) 207
ACCOUNT
operand description (asynchronous transfer) 322, 323, 334, 335
operand description (create profile) 175, 180
operand description (create remote directory) 166

ACCOUNT (cont.)
operand description (delete remote directory) 193
operand description (delete remote file) 189
operand description (display remote file attributes) 251
operand description (execute remote command) 199
operand description (modify profile) 228, 234
operand description (modify remote directory) 216
operand description (modify remote file attributes) 207
account number 322
in the remote system 166, 334, 335
accounting of file transfer requests 126
accounting record 126
ACF-2 465
ACT
explanation for output 302
request status 359
ACTIVE
request status 356
ADDRESS
explanation for output 304
addressing
partner processor 99
addressing options
Internet host name 100
ADEAC
explanation for output 302
ADM log records
display 264
ADM partner 100
ADM-CLIM
display setup 290
admission
access, remote system 166
admission profile 43, 188, 466
create 168
create (example) 185
CSV output format 399
define directory 165
delete 194
delete (example) 196
deleting 42
display 294
locking 42
modify 218
modify (example) 239
modify privilege 224
modifying 42
name specification 171
privileged 172, 466, 478
priviliging 42
time stamp 218
admission set 40, 210, 466
basic functions 172
CSV output format 385
display 254
display (example) 256
information on (example) 213
modify 209
privileged 466, 478
ADM-LOG 291
ADM-PORT 291
ADM-TRAP-SERVER 292
Advanced Encryption Standard (AES) 466
AES 273
AES (Advanced Encryption Standard) 466
AES/RSA 44, 107
alias 157
alphanumeric 466
alphanum-name (data type) 158
AMODE 466
ANSI code 466
APF authorization 81
API (Application Program Interface) 466
Application Program Interface (API) 466
asynchronous messages 108, 411
asynchronous request 28, 466
attributes
of receive files 66
authentication 48, 467
authentication check 290

authorization
access, remote system 333
login 474
LOGON 475
provide, access 106
automatic restart 30
automation 33

B
basic functions 467, 472
admission set 172
limit (IGNORE-MAX-LEVELS) 172, 224
set (MAX-LEVELS) 211
binary file 91
binary transfer 93
blank line expansion 90
block length 73
blocked records attribute 77
BS2000
access protection 106
file types 62
BS2000 file name
(DVS) syntax 54
(POSIX) syntax 55
BS2000 host 102
BYTECNT
output description 361
BYTE-COUNT
output description 360

C
CANCEL
operand description (asynchronous transfer) 341
output description 362
cancel
FT request 308
CANCELLED
request status 356
CCS 96
CCSN
output description 364
CCS-NAME
display setup 290
chaining of files, example 349
character code
define 339
Character Separated Value (CSV) 467
character set
default (operating parameter) 290
client 467
CMD
operand description (execute remote command) 198
code conversion 339
Coded Character Set (CCS) 96
CODED-CHARACTER-SET
operand description (asynchronous transfer) 326, 337
operand description (execute remote command) 198, 200
collection of files
example 347
Comma Separated Value (CSV) 467
command
abbreviate 152
execute remote 197
command execution
remote 32
with postprocessing 34
commands
remote execution 32
communication computer 467
communication controller 467, 478
composed-name (data type) 158
COMPRESS
operand description (asynchronous transfer) 338
output description 362
compressed transfer 27
compression 107, 468
computer network
open 468, 475
concatenate
libraries 139
CONDITION
operand description (asynchronous transfer) 326

connection
establishing with FTP 103
CONNECTION-LIMIT
display setup 289
connectivity 468
continuation lines 153
control
requests issued locally 279
control character attributes 77
convert
to default admission profile 222
CP1252 21
create
admission profile 168
default admission profile 171
remote directory 165
CREATION-TIME
operand description (display log records) 262
cross network connection 468
cross-domain connection 468
c-string (data type) 158
CSV format
Date data type 381
Number data type 381
String data type 381
Time data type 382
CSV output
for admission sets 385
CSV output format 37, 381
admission profile 399
admission set 385
for file attributes 383
log record 387
monitoring values 390
operating parameters 394
partner 403, 405

D

DASD (Direct Access Storage Device) 469
DASD volume 374
DATA
output description 362
data 468
data access control 71
data communication system 468
data compression 468
data conversion 21
data encoding 468
Data Encryption Standard (DES) 468
data integrity 342
data protection 469
data security 469
data set 469
data transfer
POSIX file 62
data type
alphanum-name 158
c-string 158
date 158
filename 158
integer 159
name 159
partial-name 160
text 160
time 160
x-string 160
data types 161
data types in SDF 155, 158
suffixes 155
DATA-ENCRYPTION
operand description (asynchronous transfer) 342
operand description (create profile) 184
operand description (execute remote command) 200
operand description (modify profile) 238
DATA-TYPE
operand description (asynchronous transfer) 339
operand description (execute remote command) 200
Date
data type in CSV format 381
date (data type) 158
DDICHK 273
DEACT
explanation for output 302

default account number 126
default admission profile
converting to 222
creating 171
default admission set 254, 257
default for FTP 291
default for remote administration 291
default value 152
DEFFSIZE 63
define
character code 339
direction of transfer 319
local system 312
permitted access methods 207
remote system 312
delete
admission profile 194
admission profile (example) 196
all requests 311
directories 138
FT request 308
log record 48
remote files 188
delete remote directory 191
DEL-LOG 292
DENCR 273
DES 273
DES (Data Encryption Standard) 468
DES/RSA 107
description
long output 271
DICHECK
output description 362
DICHK 273
DIERR
explanation for output 302
DIR
output description 360
Direct Access Storage Device (DASD) 469
directories
deleting 138
renaming 137
directory 469
create remote 165
delete remote 191
modify attributes 215
DIRECTORY-NAME
operand description (create remote directory) 165
operand description (delete remote directory) 191
operand description (modify remote directory) 215
display
admission profile 294
admission sets 254
admission sets (example) 256
AMD log records 264
FT partners (example) 301
FT profile (example) 297
FTAC logging records 259
information on reason codes 202
log records 259
logging records (example) 275
MAX-ADM-LEVELS 256
MAX-USER-LEVELS 256
offline log files 259
openFT instance 258
operating parameter 287
operating parameters (example) 289
partner systems 298, 305
partner systems (example) 307
remote file attributes 248
display request
global request identification 266, 357
distribution
of files (example) 348
DNS name 100
dynamic partner 290
dynamic partners 99
DYN-PART
display setup 290

E
EBCDIC 21
effects
FT profile 43
ELEMENT
operand description (asynchronous transfer) 330
emulation 469
ENC-MAND 291
ENCR 273
ENCRYPT
output description 362
encrypted file transfer 107
encryption 44
force 184
old FT versions 45
reject 184
request description data 45
user data 45
end of file
extend 339
enter
file name 52
entering a file name
specify 52
entity 473
example
chaining of files 349
collection of files 347
create admission profile 185
display FT partners 301
display FT profile 297
display logging records 275
display openFT instances 258
display operating parameters 289
display partner systems 307
display remote file attributes 252
distribution of files 348
file transfer 349, 350
file transfer for Unix system 350
file transfer using FTAC 350
file transfer with password protection 346
FTINFO for remote pre-processing 352
information on admission sets 213
local file processing 351
long output form 271
modify remote file attributes 208
NCANCEL 311
remote pre-processing 351
rename remote directory 217
short output form of FT logging records 269
execute
remote command 197
EXPANSION
admission profile 178
EXPIRATION-DATE
operand description (modify profile) 223, 224

F
FAILURE-PROCESSING 314
operand description (asynchronous transfer) 325, 337
operand description (create profile) 182
operand description (modify profile) 236
FILE
operand description (delete remote file) 188
operand description (display remote file attributes) 249
operand description (modify remote file attributes) 205
operand description (modify request queue) 241
output description 364
file
deleting 39
encrypted transfer 107
renaming 39
transfer asynchronously 312
transfer synchronously 243
file attributes 470
CSV output format 383
display 248
modify remote 204
modify remote (example) 208
modifying 39
showing 39
file format
transparent 94

file management 31, 470
interplay 39
file management function
modify in admission profile 237
file name 43, 70
specify 43
file name prefix
*ftmonitor 179, 232
file password 61
file processing 470
file structure 69
file transfer
encrypted 44
example 349, 350
for Unix system (example) 350
password protected file (example) 346
specify priority 340
using FTAC (example) 350
with postprocessing 478
file transfer request 470
start with time specification 340
file transfer request status
query 353
File Transfer, Access and Management 471
file type
BS2000 62
Unix system 89
Windows 89
z/OS 63
file utilization 133
FILE-NAME
operand description (asynchronous transfer) 320, 327
operand description (cancel request) 310
operand description (create profile) 177
operand description (display log records) 265
operand description (execute remote command) 200
operand description (modify profile) 231
operand description (query request status) 355
output description 360
selection criteria for canceling 310
filename (data type) 158
filename-prefix (data type) 159
FILE-PASSWORD
operand description (create profile) 179
operand description (modify profile) 232
files
delete remote 188
FIN
output description 359
FINISHED
request status 356
firewall processor 470
fixed record length 94
fixed-length record 470
FJCMD.TMP.OUT 146
follow-up processing 33, 34, 115, 181, 182, 233, 234, 236, 313, 470
%ELEMNAME 114
%ELEMTYP 114
%ELEMVERS 114
%FILENAME 114
%JOBCLASS 114
%PARTNER 114
%PARTNERAT 114
%RESULT 114
errors 372
in the local system 324, 325
in the remote system 336
instance 50
maximum length 115
overview 114
use of variables 325
user ID 180
variables 114, 116
with FTAM partners 115
follow-up processing request 470
front-end processor 469
FT
operand description (display log records) 263
FT administrator 471
FT log record 46
FT logging
display setup 291

FT logging record
short output form (example) 269
FT logging records 259
FT profile 40
effects 43
FT request 471, 480
cancel 308
confirmation 344
delete 308
FT system 471
FT system messages for the user 411
FT trace 471
FTAC
admission profile (privileged) 172
define directory name 192
define file name 188
fixed directory name 165
operand description (display log records) 263
password 209, 211
FTAC (File Transfer Access Control) 471
FTAC administrator 44, 471
FTAC function 38
FTAC functionality 471
FTAC log record 46
FTAC logging
display setup 291
FTAC logging function 471
FTAC logging record 202, 259
display 259
FTAC transfer admission
for FTP access 103
ftadm
protocol prefix 100
FTADM protocol 100
FTAM 471
FTAM partner
follow-up processing 115
FTAM protocol 471
FTAM-APPL 291
FTCREDIR 165
FTCREPRF 168
FTDEL 188
FTDELDIR 191
FTDELPRF 194
FTEXEC 197
FTEXECSV 271, 273
FT-FUNCTION
operand description (create profile) 183
operand description (modify profile) 237
FTHELP 202
FTINFO for remote pre-processing, example 352
FTMOD 204
FTMODADS 209
FTMODDIR 215
FTMODPRF 218
FTMODREQ 240
FT-MSP messages 108
FT-MSP return codes 108
FTP
inactive, display 291
inbound access via default FTP 103
FTP partner
addressing 100
FTP-PORT 291
FTR messages 413
FTSCOPY 243
FTSHW 248
FTSHWADS 254
FTSHWINS 258
FTSHWLOG 259
FTSHWMON 276
CSV format 390
FTSHWOPT 287
FTSHWPRF 222, 294
example 297
FTSHWPTN 298
Beispiel 301
FTSHWRGE 305
CSV output 405

G

gateway 472
gateway processor 472
Generalized Trace Facility (GTF) 472
global request identification 362
display request 266, 357

GLOB-ID
output description 362
GTF (Generalized Trace Facility) 472

H
heterogeneous
computer systems 21
link 51
network 24, 472
HOLD
output description 359
request status 357
homogeneous link 51
homogeneous network 24, 472
HOST-NAME 291

I
IBF 256
IBP 256
IBR 256
IBS 256
IDENTIFICATION
Einstellung anzeigen 292
explanation for output 304
identification 472
of a FT request 344
identify
request 344
IDREJ
explanation for output 302
IEBCOPY 472
IEBGENER 472
IEBPTPCH 473
IGNORE-MAX-LEVELS
operand description (create profile) 172
operand description (modify profile) 224
inbound
file management 39, 473
follow-up processing 39, 473
receive 39, 473
request 473
requests 27
send 39, 473
submission 473
inbound access
FTP 103
inbound file management 174, 227
inbound follow-up processing 174
inbound processing 227
inbound receive 174, 212, 226
inbound send 174, 212, 226
INBOUND-FILEMANAGEMENT 256
INBOUND-MANAGEMENT
operand description (create profile) 174
operand description (modify admission set) 213
operand description (modify profile) 227
INBOUND-PROCESSING 256
operand description (create profile) 174
operand description (modify admission set) 213
operand description (modify profile) 227
INBOUND-RECEIVE 256
operand description (create profile) 174
operand description (modify admission set) 212
operand description (modify profile) 226
INBOUND-SEND 256
operand description (create profile) 174
operand description (modify admission set) 212
operand description (modify profile) 226
INFORMATION
operand description (display log records) 268
operand description (display partners) 300
operand description (display profiles) 295
operand description (display remote file attributes) 251
operand description (query request status) 357
operand description (showing monitoring data) 277
information
getting on operating parameters 276
INI
output description 359

INITIATOR
operand description (cancel request) 310
operand description (create profile) 176
operand description (display log records) 264
operand description (modify profile) 229
operand description (query request status) 355
output description 361
initiator 473
instance 50, 473, 476
preprocessing, postprocessing follow-up processing 50
setting 50, 139
instance ID 473
instance identification 48
integer (data type) 159
integrity 473
Interactive Problem Control System (IPCS) 473
Internet host name
addressing options 100
Internet Protocol (IP) 483
interoperability 473
interplay
file management 39
IPCS (Interactive Problem Control System 473
IPv4 address 100
ISAM file
transferring to a foreign system 95
ISO 8859 21
ISO reference model 473
ISO/OSI reference model 473
ISPF 139, 140, 474
ISPF/PDF 140, 474

J

job 474
transfer 474

K

key pair set 49
KEY-LEN
display setup 290
keyword
form 153
operands 152

L

LAN (Local Area Network) 474
LAUTH 273
explanation for output 302
LAUTH2 273
LAYOUT
operand description (display admission sets) 255
operand description (display log records) 269
operand description (display operating parameters) 288
operand description (display partners (FTAC) 306
operand description (display partners) 299
operand description (display profiles) 296
operand description (display remote file attributes) 252
operand description (query request status) 358
operand description (showing monitoring data) 278
length
RSA key 290
LIBRARY
operand description (asynchronous transfer) 330
library 474
concatenating 139
name in the remote system 330
library member 474
name in the remote system 330
lifetime
request 28
limit
basic functions (IGNORE-MAX-LEVELS) 172
limit basic functions (IGNORE-MAX-LEVELS) 224

link
heterogeneous 51
homogeneous 51
LISTING
operand description (asynchronous transfer) 325
LOC
explanation for output 303
output description 363
Local Area Network (LAN) 474
local file processing, example 351
local system 320, 474
LOCAL SYSTEM NAME
Einstellung anzeigen 292
LOCAL-PARAMETER
definition of local system 312
operand description (asynchronous transfer) 320
LOCK
output description 359
log files
output names 268
log record
display 259
log records 474
CSV output format 387
repeat output 268
logging 46
display setup 291
postprocessing 47
preprocessing 47
logging function 474
logging record 202, 371
LOGGING-ID
operand description (display log records) 261
Logical Unit (LU) 474
login
FTP 103
login admission 40
login authorization 474
LOGON authorization 175, 228, 475
LOGON procedure 140
long form 152
long output
description 271
long output form
example 271
lowercase 153
lowercase letters 153
LU (logical unit) 474
LUNK
explanation for output 302

M

macro
OPENFT 365
mainframe 475
managed file transfer 19
mandatory parameter 313
MAX-ADM-LEVELS 211, 256
description of output fields 256
MAXALLOC 64
maximum
lifetime of a request 290
number of asynchronous administration requests 290
number of connections 289
number of FT requests 290
number of tasks 289
maximum record length 104
MAX-LEVELS
operand description (modify admission set) 211
MAX-PARTNER-LEVEL
operand description (create profile) 177
operand description (modify profile)F) 230
MAX-REQUEST-LIFETIME 290
display setup 290
MAX-USER-LEVELS 211, 256
description of output fields 256
member list 105
menu interface for FT users 139
message code 411
messages
FTR 413
metasyntax 156
of SDF 155

migrated file
transfer 312
modify
admission profile 218
admission profile (example) 239
admission set 209
file management function in admission profile 237
privilege in admission profile 224
remote directory attributes 215
remote file attributes 204
remote file attributes (example) 208
request queue 240
monitoring
deactivated for partners 279
profile for 179, 232
showing setting 293
monitoring data
show 276
multivolume file 65

N

NAME
explanation for output 301
operand description (create profile) 171
operand description (delete profile) 194
operand description (display profiles) 294
operand description (display remote file attributes) 250
operand description (modify profile) 221
operand description (showing monitoring data) 276
name
remote system 312
specification for admission profile 171
name (data type) 159
NCANCEL 308
cancel file transfer 308
example 311
NCOPY 312
full form 316
NCP (Network Control Program) 475
NetMaster 475
NetView 475
network
definition 24
heterogeneous 24, 472
homogeneous 24, 472
Network Control Program (NCP) 475
network description file 475
network management 24
networks
openFT support 24
NEW-NAME
operand description (modify profile) 222
operand description (modify remote directory) 217
operand description (modify remote file attributes) 207
NEW-PASSWORD
operand description (modify admission set) 211
NOCON
explanation for output 302
NOKEY
explanation for output 302
notational conventions 15
notational conventions for SDF 155
NSTATUS 353
output in CSV format 406
NUMBER
operand description (display log records) 267
Number
data type in CSV format 381
number
display maximum of transport connections 289
number (data type) 159
number of directory blocks 78

O

object 475
OBR 256
OBS 256

offline log file
selection according to date 267
selection according to name 267
specify number 267
offline log files
display 259
offline log records
view 266
old FT versions
encryption 45
open computer network 468
openEdition directories 128
openEdition file 63
%UNIQUE 52
syntax 59
transfer 65
WRITE-MODE 67
openFT
partner 476
openFT add-on products 23
openFT instance 139
display 258
display (example) 258
openFT instances 50
openFT partner 24
addressing 100
openFT protocol
addressing with 100
openFT protocols 24, 476
openFT return codes 163
openFT-AC 218
OPENFT-APPL
display setup 291
openFT-FTAM 476
operand 152
operand value
constant 152
introductory 152
operating parameter
display 287
display (example) 289
outputting 276
operating parameters 476
CSV output format 394
OSI reference model 473
outbound
receive 39, 476
request 476
requests 27
send 39, 476
submission 476
outbound encryption
activate 218
outbound receive 173, 212, 226
outbound request 240
outbound send 173, 211, 225
OUTBOUND-RECEIVE 256
operand description (create profile) 173
operand description (modify admission set) 212
operand description (modify profile) 226
OUTBOUND-SEND 256
operand description (create profile) 173
operand description (modify admission set) 211
operand description (modify profile) 225
OUTPUT
operand description (asynchronous transfer) 326
operand description (display admission sets) 255
operand description (display log records) 269
operand description (display operating parameters) 288
operand description (display partners (FTAC) 306
operand description (display partners) 299
operand description (display profiles) 296
operand description (display remote file attributes) 251
operand description (execute remote command) 200
operand description (query request status) 357
operand description (showing monitoring data) 278

output fields
description (show log record) 269
description (show operating parameters) 289
output in CSV format 37
admission sets 385
FTSHW 383
FTSHWLOG 387
FTSHWMON 390
FTSHWOPT 394
FTSHWPTN 403
FTSHWRGE 405
NSTATUS 406
overwrite
receive file 338
OWNER
output description 362
owner 476
of a FT request 308
of FT request 476
OWNER-IDENTIFICATION 310
OWNER-IDENTIFICATION
operand description (cancel request) 310
operand description (delete profile) 195
operand description (display log records) 262
operand description (display profiles) 295
operand description (modify profile) 222
operand description (modify request queue) 241
operand description (query request status) 355

P

PAM file
fetching from a foreign system 95
transferring to a foreign system 95
partial-filename (data type) 160
partitioned data set extended 477
partitioned organized data set 477
PARTNER
operand description (asynchronous transfer) 319
operand description (cancel request) 310
operand description (create profile) 177
operand description (create remote directory) 165
operand description (delete remote directory) 191
operand description (delete remote file) 188
operand description (display log records) 264
operand description (display partners) 299
operand description (display remote file attributes) 249
operand description (execute remote command) 198
operand description (modify profile) 230
operand description (modify remote directory) 215
operand description (modify remote file attributes) 205
operand description (modify request queue) 241
operand description (query request status) 355
output description 359, 362
partner
CSV output format 403
partner address 99
partner list 99
partner name 99
partner processor
addressing 99
partner system 312, 477
display 298, 305
display (example) 307
specify 43
PARTNER-CHECK
display setup 290
PARTNER-STATE
operand description (query request status) 355
PASSWORD
operand description (asynchronous transfer) 321, 322, 323, 332, 334, 335
operand description (create profile) 171, 176, 180

PASSWORD (cont.)
operand description (create remote directory) 166, 167
operand description (delete profile) 195
operand description (delete remote directory) 192, 193
operand description (delete remote file) 189
operand description (display remote file attributes) 250, 251
operand description (execute remote command) 199
operand description (modify admission set) 210
operand description (modify profile) 221, 229, 234
operand description (modify remote directory) 216
operand description (modify remote file attributes) 206, 207
password 209, 254, 477
access, remote system 167
P-CHK
explanation for output 303
PDSE data set 477
PDSE member 58
PDSE member, file consistency 105
physical sequential data set 477
Physical Unit (PU) 477
PO data set 477
PO member 58
PO member, file consistency 105
POLLING
operand description (showing monitoring data) 277
polling
cancel (log records) 268
log records 268
polling interval
log records 268
polling log records
number of repetitions 268
port number 291, 477
default for openFT 291
partner host 100
Portable Open System Interface (POSIX) 477
positional form 153
positional operands 152
POSIX (Portable Open System Interface) 477
POSIX file
file format during transfer 62
posix filename (data type) 59
posix path name (data type) 55
posix pathname (data type) 59
post-processing
logging record 271, 273
set up 168
postprocessing 33, 373, 478
function 34
instance 50
logging 47
previous FT versions 34
post-processing command
local 320, 329
postprocessing commands
local 112
PREFIX
operand description (create profile) 181, 182
operand description (modify profile) 235, 236
prefix
specify for file name 43
specify for follow-up processing 44
pre-processing
logging record 271, 273
set up 168
preprocessing 33, 34, 373, 478
description 111
instance 50
logging 47
pre-processing command
local 320
remote system 329
preprocessing commands
local 111
preprocessor 478
presentation selector
partner host 101
primary allocation 63
PRIMARY OPTION MENU 143

PRIO
output description 361
PRIORITY
operand description (asynchronous transfer) 340
operand description (modify request queue) 242
priority 340
partners 29
specify for file transfer 340
priority control 29
private key 478
private volumes 65
PRIVILEGED 218
operand description (create profile) 172
operand description (modify admission set) 211
operand description (modify profile) 224
privileged admission profile 478
privileged admission set 466, 478
procedure 478
procedure call
postprocessing 34
processing
prohibited 43
specified 43
PROCESSING-ADMISSION
operand description (asynchronous transfer) 323, 334
operand description (create profile) 179
operand description (modify profile) 233
specify user ID for follow-up processing 314
PROCESS-LIMIT
display setup 289
product range
openFT 22
profile 478
program call
postprocessing 34
preprocessing 34
program interface for the FT user 365
program interfaces 36
prohibited processing 43
PROTECT command 321
protocol 479
PS data set 477
PS dataset 57
PU (Physical Unit) 477
public key 479
PW 256

Q

query
status of file transfer request 353
QUEUE-POSITION
operand description (modify request queue) 241
quotes 152

R

RACF 136, 479
RAUTH 273
explanation for output 302
RAUTH2 273
read password
receive file 321, 332
send file 321
reason code 202
display information 202
REASON-CODE
operand description (display log records) 266
receive file 479
overwrite 338
read password 321, 332
write password 321, 332
receive system 319, 479
record 479
record format 72
record length 74, 104, 470, 484
record-by-record transfer 94
RECORD-FORMAT
operand description (asynchronous transfer) 342
RECORD-SIZE
operand description (asynchronous transfer) 342

RECORD-TYPE
operand description (display log records) 263
relay program 479
REM
explanation for output 303
output description 364
remote command execution 32
remote directory
rename (example) 217
remote file attribute
display (example) 252
remote pre-processing, example 351
remote system 327, 480
identification of user 251
REMOTE-PARAMETER
definition of remote system 312
operand description (asynchronous transfer) 327
rename
directories 137
request 480
asynchronous 28, 466
identify 344
lifetime 28
owner 308
priority 29
synchronous 28, 243, 483
request confirmation 108
request description data
encrypting 20
Request for Comments (RFC) 481
request ID 480
request identification 480
request information
about FT requests 353
request lifetime 290
request management 480
request number 480
request queue 29, 480
modify 240
request rejection 108
request storage 480
REQUEST-LIMIT
display setup 290
resources 480
responder 480
restart 480
automatic 30
pre- and post-processing 321
pre-/post-processing 329
restart capability
postprocessing 113
restart point 480
restriction
transfer direction 43
write mode (FT profile) 44
result list 325, 481
result lists 110
result message 313, 373
return code
new variant 368
old variant 368
REXX 481
RFC (Request for Comments) 481
RFC1006 481
Rivest-Shamir-Adleman procedure 481
ROUTING
explanation for output 304
RSA 273
RSA key, length 290
RSA procedure 481
RSA/AES 44, 107
RSA/DES 107
RUNK
explanation for output 302

S

SECLEV
explanation for output 303
secondary allocation 63
Secure FTP 481
secure operation 38
security level 177, 211, 481
default value 290
SECURITY-LEVEL
display setup 290

SELECT
operand description (cancel request) 310
operand description (display log records) 261
operand description (modify request queue) 241
operand description (query request status) 354
selection criteria
for FT requests 310
for FT requests to be canceled 310
for outbound requests to be modified 241
SELECT-PARAMETER
operand description (delete profile) 195
operand description (display admission sets) 255
operand description (display partners (FTAC) 306
operand description (display profiles) 295
operand description (modify admission set) 210
operand description (modify profile) 221
send file 481
binary transfer 339
read password 321
send system 319
sending system 481
server 481
service 482
session 482
session selector 482
partner computer 101
set up
post-processing 168
pre-processing 168
setting an instance 50, 139
setup
transfer admission 171
short form 152
show
monitoring data 276
Siemens protocols 24
Simple Network Management Protocol (SNMP) 482
SMF 126
SMF (System Management Facility) 482
SMP/E (System Modification Program/ Extended) 482
SN77309 24
SN77312 24
SNA LU name 100
SNA network 482
SNMP 24
SNMP (Simple Network Management Protocol) 482
spanned records attribute 77
special form (*DELETE) 115
specify 43
file transfer request 51
partner processor 99
partner systems 43
prefix for file name 43
prefix for follow-up processing 44
processing 43
syntax rules 104
transfer admission 102
user ID for follow-up processing 314
standard admission profile 43
standard admission set 482
START 340
operand description (asynchronous transfer) 340
output description 362
start of file 338
start of the file transfer 340
STARTED
display setup 289
STATE
explanation for output 301
operand description (display partners) 300
operand description (query request status) 356
output description 359, 361
status
of FT request 356
storage allocation 70
storage space 374

String
data type in CSV format 381
string 482
successful file transfer 325
SUCCESS-PROCESSING 314
operand description (asynchronous transfer) 325, 336
operand description (create profile) 181
operand description (modify profile) 234
SUFFIX
operand description (create profile) 181, 183
operand description (modify profile) 235, 236
suffixes for data types 155, 161
SUSP
output description 359
synchronous request 28, 243, 483
syntax
BS2000 327
BS2000 file name (DVS) 54
BS2000 file name (POSIX) 55
MSP 327
Unix system file name 56
Windows file name 56
z/OS file name 57
syntax check 327
syntax rules
specify 104
Sysplex composite 50
system 483
local 474, 483
remote 480, 483
System Management Facility SMF) 482
System Modification Program/Extended (SMP/E) 482
SYSUT1 329

T

TABULATOR
operand description (asynchronous transfer) 343
tabulator expansion 90
task 483
TCP/IP 483
text (data type) 160
text file
transfer 339
text format 89
data conversion 21
text transfer 93
Time
data type in CSV format 382
time (data type) 160
time stamp
updating on admission profile 218
TNS name 100
TNSTCPIP 100
Top Secret 483
total length of command string 369
TRACE
display setup 293
explanation for output 303
TRANS
output description 361
TRANS-ADM
output description 364
transfer
as a text file 339
binary 339
encrypted 107
file asynchronously 312
file synchronously 243
in binary format 93
in compressed form 338
in text format 93
in user format 93
migrated file 312
record-by-record 94
transparent format 94
transfer admission 221, 222, 294, 321, 371, 483
file transfer request 43
FTAC 40
setup 171
specify 102
transfer direction 229
restriction 43

transfer file
DVS file 62
file name syntax 62
library element 62
PLAM library 62
POSIX file 62
transfer ID 240, 309, 354
TRANSFER-ADMISSION 218
operand description (asynchronous transfer) 321, 333
operand description (create profile) 171
operand description (create remote directory) 166
operand description (delete profile) 195
operand description (delete remote directory) 192
operand description (delete remote file) 189
operand description (display profiles) 295
operand description (display remote file attributes) 250
operand description (execute remote command) 199
operand description (modify profile) 221, 222, 223
operand description (modify remote directory) 216
operand description (modify remote file attributes) 206
TRANSFER-DIRECTION
operand description (asynchronous transfer) 319
operand description (create profile) 176
operand description (modify profile) 229
TRANSFER-ID
operand description (cancel request) 309
operand description (modify request queue) 240
operand description (query request status) 354
output description 361
request identification 309
TRANS-ID
output description 359
Transmission Control Protocol (TCP) 483
TRANSP
output description 362
TRANSPARENT
operand description (asynchronous transfer) 340
transparent file format 94
transparent format
transfer 94
transport connection 483
transport connections
display maximum number 289
transport layer 484
transport protocol 484
transport selector 484
partner host 100
transport system 24, 484
transport unit
maximum size 290
TRANSPORT-UNIT-SIZE
display setup 290
TRAP
display setup 292
T-selector 484
TYPE
operand description (asynchronous transfer) 331
types
follow-up processing 114

U

umlauts
data conversion 21
UNC names 56
Unicode 21
Unix system
access protection 105
file name, syntax 56
file types 89
UNIX(TM) 484
uppercase 153
uppercase letters 153
uppercase/lowercase notation 143
USAGE
operand description (modify profile) 223, 224

user data
encryption 45
user format
transfer 93
user ID 210, 233, 254, 484
admission profile 180
delete admission profile 195
in the remote system 166
user identification 484
in the remote system 335
USER-ADMISSION
operand description (create profile) 175
operand description (modify profile) 227
user-generated result information 110
USER-IDENTIFICATION
operand description (asynchronous transfer) 322, 323, 334, 335
operand description (create profile) 175, 180
operand description (create remote directory) 166
operand description (delete remote directory) 192
operand description (delete remote file) 189
operand description (display admission sets) 254
operand description (display partners (FTAC) 306
operand description (display remote file attributes) 251
operand description (execute remote command) 199
operand description (modify admission set) 210
operand description (modify profile) 228, 233
operand description (modify remote directory) 216
operand description (modify remote file attributes) 206
USER-INFORMATION
operand description (create profile) 184
operand description (modify profile) 238

V

VALID
operand description (modify profile) 223, 224
variable-length record 484
variables
follow-up processing 114, 116
VERSION
operand description (asynchronous transfer) 331
volume for receive files 70
volume for result list files 110
volumes 65, 70, 110
VSAM 485
VSAM file 58
VTAM 485

W

WAIT 375
output description 359
WAN (Wide Area Network) 485
Wide Area Network (WAN) 485
wildcards
partners in ftshwl 265
Windows
access protection 105
file types 89
Windows file name
syntax 56
WRITE
output description 361
write mode
restriction 44
write password
receive file 321, 332
WRITE-MODE
operand description (asynchronous transfer) 338
operand description (create profile) 183
operand description (modify profile) 237

X

XMIT 94
x-string (data type) 160

Z

z/OS
    file name, syntax 57
    file type 63
z/OS UNIX System Services 59